TUGAS AKHIR - SM 141501

PENGKLASTERAN LAPORAN TUGAS AKHIR BERDASARKAN ABSTRAK MENGGUNAKAN METODE *RAPID AUTOMATIC KEYPHRASE EXTRACTION* DAN *AVERAGE LINKAGE HIERARCHICAL CLUSTERING*

MEGA FATMAWATI
NRP 1213 100 005

Dosen Pembimbing
Alvida Mustika Rukmi, S.Si, M.Si
Drs. Soetrisno, MI.Komp.

DEPARTEMEN MATEMATIKA
Fakultas Matematika dan Ilmu Pengetahuan Alam
Institut Teknologi Sepuluh Nopember
Surabaya 2017

TUGAS AKHIR – SM141501

PENGKLASTERAN LAPORAN TUGAS AKHIR BERDASARKAN ABSTRAK MENGGUNAKAN METODE *RAPID AUTOMATIC KEYPHRASE EXTRACTION* DAN *AVERAGE LINKAGE HIERARCHICAL CLUSTERING*

MEGA FATMAWATI
NRP 1213 100 005

Dosen Pembimbing
Alvida Mustika Rukmi, S.Si, M.Si
Drs. Soetrisno, MI.Komp.

DEPARTEMEN MATEMATIKA
Fakultas Matematika dan Ilmu Pengetahuan Alam
Institut Teknologi Sepuluh Nopember
Surabaya 2017

FINAL PROJECT – SM141501

FINAL PROJECT REPORTS CLUSTERING BASED ON ABSTRACT USING RAPID AUTOMATIC KEYPHRASE EXTRACTION AND AVERAGE LINKAGE HIERARCHICAL CLUSTERING METHODS

MEGA FATMAWATI
NRP 1213 100 005

Supervisors
Alvida Mustika Rukmi, S.Si, M.Si
Drs. Soetrisno, MI.Komp.

DEPARTMENT OF MATHEMATICS
Faculty of Mathematics and Natural Sciences
Sepuluh Nopember Institute of Technology
Surabaya 2017

LEMBAR PENGESAHAN

PENGKLASTERAN LAPORAN TUGAS AKHIR BERDASARKAN ABSTRAK MENGGUNAKAN METODE *RAPID AUTOMATIC KEYPHRASE EXTRACTION* DAN *AVERAGE LINKAGE HIERARCHICAL CLUSTERING*

FINAL PROJECT CLUSTERING REPORTS BASED ON ABSTRACT USING RAPID AUTOMATIC KEYPHRASE EXTRACTION AND AVERAGE LINKAGE HIERARCHICAL CLUSTERING METHODS

TUGAS AKHIR

Diajukan untuk memenuhi salah satu syarat
Untuk memperoleh gelar Sarjana Sains
Pada bidang studi Ilmu Komputer
Program Studi S-1 Departemen Matematika
Fakultas Matematika dan Ilmu Pengetahuan Alam
Institut Teknologi Sepuluh Nopember Surabaya

Oleh :
MEGA FATMAWATI
NRP. 1213 100 005

Menyetujui Dosen Pembimbing II	Menyetujui Dosen Pembimbing I
Drs. Soetrisno, MI.Komp.	Alvida Mustika Rukmi, S.Si, M.Si
NIP. 19571103 198603 1 003	NIP. 19720715 199802 2 001

Mengetahui
Kepala Departemen Matematika
FMIPA ITS

Dr. Imam Mukhlash, S.Si, MT
NIP. 19700831 199403 1 003
Surabaya, Juli 2017

PENGKLASTERAN LAPORAN TUGAS AKHIR BERDASARKAN ABSTRAK MENGGUNAKAN METODE *RAPID AUTOMATIC KEYPHRASE EXTRACTION* DAN *AVERAGE LINKAGE HIERARCHICAL CLUSTERING*

Nama Mahasiswa	**: Mega Fatmawati**
NRP	**: 1213 100 005**
Departemen	**: Matematika FMIPA-ITS**
Pembimbing	**: 1. Alvida Mustika Rukmi, S.Si, M.Si**
	2. Drs. Soetrisno, MI.Komp.

Abstrak

Tugas Akhir merupakan salah satu syarat wajib mahasiswa S1 ITS untuk mendapatkan gelar sarjana. Setiap mahasiswa tingkat akhir seringkali kesulitan dalam menentukan pokok pembahasan atau topik apa yang akan dibahas di laporan Tugas Akhir. Oleh karena itu, pada penelitian ini disajikan pengklasteran laporan Tugas Akhir berdasarkan abstrak. Metode *Rapid Automatic Keyphrase Extraction* (RAKE) digunakan untuk mengekstraksi kata penting yang ada di abstrak laporan Tugas Akhir mahasiswa ITS. Parameter jumlah kata peting pada RAKE mempengaruhi kualitas pengklasteran dokumen. Metode *Average Linkage Hierarchical Clustering* digunakan untuk pengklasteran laporan Tugas Akhir mahasiswa ITS. Hasil pengklasteran berdasarkan jumlah kata penting dapat memberikan informasi mengenai topik – topik dalam bentuk *cluster – cluster.* Pada Tugas Akhir ini uji coba dilakukan terhadap 3 data departemen yaitu Matematika, Fisika dan Teknik Perkapalan. Berdasarkan hasil uji coba, pengklasteran terbaik dilakukan dengan menggunakan 2 kata penting.

Kata Kunci : Rapid Automatic Keyphrase Extraction, Pengklasteran, Tugas Akhir, Average Linkage Hierarchical Clustering

FINAL PROJECT REPORTS CLUSTERING BASED ON ABSTRACT USING RAPID AUTOMATIC KEYPHRASE EXTRACTION AND AVERAGE LINKAGE HIERARCHICAL CLUSTERING METHODS

Name	**: Mega Fatmawati**
NRP	**: 1213 100 005**
Department	**: Mathematics FMIPA-ITS**
Supervisors	**: 1. Alvida Mustika Rukmi, S.Si, M.Si**
	2. Drs. Soetrisno, MI.Komp.

Abstract

The final project is one of the conditions of the compulsory undergraduate students of Institut Teknologi Sepuluh Nopember (ITS) to get a degree. Every student who wants to work the final assignment usually difficult to determine the topics will be covered in the final project reports.. Therefore, in this study presented clustering final project reports based on abstract. A method of Rapid Automatic Keyphrase Extraction (RAKE) is used to extraction the important words that exist in ITS student final project abstracts. Average Linkage method of Hierarchical Clustering are used to clustering reports student final project ITS. Clustering results based on word count can provide information on important topics – topics in a cluster. In this final project trials conducted against the 3 departments namely mathematics, physics and marine engineering. Based on the results of the experiment, best clustering using 2 important words.

Keywords : Rapid Automatic Keyphrase Extraction, Clustering, Final Project, Average Linkage Hierarchical Clustering.

KATA PENGANTAR

Segala puji syukur penulis panjatkan ke hadirat Allah SWT, karena dengan ridlo-Nya penulis dapat menyelesaikan Tugas Akhir yang berjudul

"PENGKLASTERAN LAPORAN TUGAS AKHIR BERDASARKAN ABSTRAK MENGGUNAKAN METODE *RAPID AUTOMATIC KEYPHRASE EXTRACTION* DAN *AVERAGE LINKAGE HIERARCHICAL CLUSTERING*"

yang merupakan salah satu persyaratan akademis dalam menyelesaikan Program Sarjana Departemen Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Institut Teknologi Sepuluh Nopember Surabaya.

Tugas Akhir ini dapat diselesaikan dengan baik berkat kerja sama, bantuan, dan dukungan dari banyak pihak. Sehubungan dengan hal tersebut, penulis ingin mengucapkan terima kasih kepada :

1. Bapak Dr. Imam Mukhlash, S.Si, MT selaku Kepala Departemen Matematika ITS.
2. Ibu Dra. Sri Suprapti Hartatiati, M.Si selaku Dosen Wali yang telah memberikan arahan akademik selama penulis menempuh pendidikan di Departemen Matematika ITS.
3. Ibu Alvida Mustika Rukmi, S.Si, M.Si selaku Dosen Pembimbing I yang telah memberikan bimbingan dan motivasi kepada penulis dalam mengerjakan Tugas Akhir ini sehingga dapat terselesaikan dengan baik.
4. Bapak Drs. Soetrisno, MI.Komp. selaku Dosen Pembimbing II yang telah memberikan bimbingan dan motivasi kepada penulis dalam mengerjakan Tugas Akhir ini sehingga dapat terselesaikan dengan baik.
5. Dr. Didik Khusnul Arif, S.Si, M.Si selaku Ketua Program Studi S1 Departemen Matematika ITS.
6. Drs. Iis Herisman, M.Si selaku Sekretaris Program Studi S1 Departemen Matematika ITS.

7. Seluruh jajaran dosen dan staf Departemen Matematika ITS.
8. Keluarga tercinta yang senantiasa memberikan dukungan dan do'a yang tak terhingga.
9. Teman-teman angkatan 2013 yang saling mendukung dan memotivasi.
10. Semua pihak yang tak bisa penulis sebutkan satu-persatu, terima kasih telah membantu sampai terselesaikannya Tugas Akhir ini.

Penulis menyadari bahwa Tugas Akhir ini masih jauh dari kesempurnaan. Oleh karena itu, penulis mengharapkan kritik dan saran dari pembaca. Akhir kata, semoga Tugas Akhir ini dapat bermanfaat bagi semua pihak yang berkepentingan.

Surabaya, Juli 2017

Penulis

Special thanks to

Keberhasilan penulisan Tugas Akhir ini tidak lepas dari bantuan dan dukungan dari orang-orang terdekat penulis. Oleh sebab itu, penulis mengucapkan terimakasih dan apresiasi secara khusus kepada:

1. Bapak Hariyadi dan Ibu Sri Suati, kedua orang tua penulis yang selalu memberikan doa terbaik, kasih sayang, dukungan, motivasi, dan nasehat kepada penulis.
2. Adi Purwanto, kakak penulis yang selalu memberikan semangat dan dukungan serta kepercayaan kepada penulis.
3. Ciptya Rahma Almira, sahabat penulis yang selalu mendengarkan keluh kesah penulis, memberikan dukungan, semangat, motivasi, waktu dan kecerian kepada penulis.
4. Neni Imro'atus Sholikhah, Nurma Arika Widya Yoga, Siti Nur Diana, Eries Bagita Jayanti, teman teman bimbingan penulis yang selalu memberikan dukungan dan semangat kepada penulis ketika penulis *low* motivasi, memberikan bantuan kepada penulis ketika kesusahan dalam pengerjaan dan mengijinkan penulis menginap.
5. Fedric Fernando, Hartanto Setiawan, Gina Faaizatud Dini, Metta Andriana yang membantu penulis dalam mengerjakan program.
6. Ivan Oktaviano, Putri Saraswati, Siti Nur Afifah, Retno Palupi, Ayu Enitasari Aprilia, Niken Ratna Wahyu Ningrum, Ayu Risanti Yuniar, Lisa Anisa, Gery Dias Claudio, Ina Nur Solihah, Jessica Rahma Prillantika, Melynda Sylvia Dewi, Yenny Triningsih, Muslimatun Nadhifa, Dinda Ulima, Sinar Dwi amutu, Frikha Anggita, Nastitie, Dinan Farhana dan teman – teman seperjuangan Matematika 2013 yang lain yang tidak dapat penulis sebutkan satu persatu yang selalu memberikan doa,

dukungan, motivasi, serta bantuan kepada penulis selama ini.

7. Mustikowati, Mbak Annisa, Achmad Arianto dan fikri yang selalu memberikan doa, semangat, dukungan dan bantuan kepada penulis.
8. Putranti Kusumawardani, Reyesta Hajar, Friska Khistiningtyas, Al Wafdah Lazuardian sahabat penulis dari SMA yang selalu memberikan motivasi, doa dan dukungan kepada penulis.
9. Mbak devi, mbak cindy serta teman teman kos penulis yang selalu memberikan semangat serta dukungan kepada penulis.
10. Semua pihak yang tak bisa penulis sebutkan satu-persatu, terima kasih telah membantu sampai terselesaikannya Tugas Akhir ini.

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB I
PENDAHULUAN

Pada bab ini dibahas mengenai latar belakang yang mendasari penulisan Tugas Akhir ini. Di dalamnya mencakup identifikasi permasalahan pada topik Tugas Akhir kemudian dirumuskan menjadi permasalahan yang diberikan batasan-batasan dalam pembahasan pada Tugas Akhir ini.

1.1 Latar Belakang

Tugas Akhir merupakan salah satu syarat wajib mahasiswa S1 Institut Teknologi Sepuluh Nopember (ITS) untuk mendapatkan gelar sarjana. Tugas Akhir dimaksudkan untuk melatih mahasiswa melakukan penelitian ilmiah secara mandiri dan menyusun karya ilmiah yang berkualitas. Tujuan diwajibkannya mahasiswa mengambil mata kuliah Tugas Akhir adalah agar mahasiswa memiliki pemahaman yang baik tentang standar kualitas karya ilmiah di tingkat S1, memiliki kemampuan bekerja mandiri, memiliki kemampuan berargumentasi secara ilmiah, memiliki kebiasaan bekerja secara sistematis dan tepat waktu, memiliki sifat terbuka, jujur, kritis dan bertanggungjawab, serta memiliki kemampuan mengembangkan imajinasi, sikap kreatif dan inovatif [1]. Setiap mahasiswa menentukan pokok pembahasan atau topik terlebih dahulu sebelum mengerjakan Tugas Akhir agar fokus terhadap penelitiannya. Terdapat berbagai cara agar setiap mahasiswa mendapatkan topik tugas akhir yang sesuai dengan keinginannya, seperti membaca jurnal penelitian dalam *e-journal*, mengikuti penelitian yang dilakukan oleh dosen, membaca laporan Tugas Akhir yang pernah dibuat, mengamati permasalahan yang ada di sekitar, dll. Namun pada kenyataannya, mencari topik Tugas Akhir bukanlah hal yang mudah. Hal ini dibuktikan dengan masih banyaknya mahasiswa ITS yang kesulitan untuk memulai mengerjakan

laporan Tugas Akhir karena belum mendapatkan topik yang sesuai dengan keinginan [2].

Informasi mengenai laporan Tugas Akhir yang pernah dibuat dapat diperoleh di perpustakaan ITS. Tahun 2012 sampai 2016, perpustakaan ITS telah mengarsipkan sebanyak 6.916 laporan Tugas Akhir mahasiswa dari setiap departemen yang ada di ITS. Artinya terdapat 6.916 topik yang telah diteliti. Akan tetapi, banyaknya laporan Tugas Akhir yang telah diarsipkan mengakibatkan mahasiswa ITS kesulitan dalam mengetahui isi setiap dokumen. Untuk mempermudah menentukan isi setiap dokumen, diperlukan kata penting yang mampu mewakili isi dokumen. Kata penting tersebut dapat diperoleh dari ringkasan dan kata kunci abstrak.Untuk mendapatkan kata penting tersebut diperlukan ekstraksi kandidat kata penting dari ringkasan dan kata kunci abstrak. Ekstraksi kata penting secara otomatis dapat dilakukan dengan menggunakan metode *Rapid Automatic Keyphrase Extraction* (RAKE). Salah satu metode *clustering* yang sering digunakan adalah metode *Average Linkage Hierarchical Clustering*. Metode ini relatif yang terbaik dari metode – metode *hierarchical* lainnya karena proses *clustering*nya didasarkan pada jarak rata – rata antar obyeknya [3].

Berdasarkan latar belakang tersebut, pada Tugas Akhir ini penulis menggunakan metode *Rapid Automatic Keyphrase Extraction* (RAKE) untuk ekstraksi kata penting dan metode *Average Linkage Hierarchical Clustering* untuk pengklasteran topik Tugas Akhir mahasiswa ITS. Data yang digunakan adalah abstrak Bahasa Indonesia dari laporan Tugas Akhir Mahasiswa ITS.

1.2 Rumusan Masalah

Berdasarkan latar belakang tersebut, dapat dirumuskan permasalahan dalam Tugas Akhir ini adalah sebagai berikut :

1. Bagaimana menentukan kata penting dalam abstrak Tugas Akhir mahasiswa ITS menggunakan *Rapid Automatic Keyphrase Extraction* (RAKE) ?
2. Bagaimana membuat pengklasteran topik Tugas Akhir mahasiswa ITS berdasarkan abstrak menggunakan *Average Linkage Hierarchical Clustering* ?

1.3 Batasan Masalah

Pada penelitian ini, penulis membuat batasan masalah sebagai berikut :

1. Data yang digunakan adalah database laporan Tugas Akhir mahasiswa ITS yang berada di *digital library* ITS pada tahun 2012 sampai dengan tahun 2016.
2. Pengklasteran dokumen hanya dilakukan di departemen Matematika, Fisika dan Teknik Perkapalan.
3. Format data adalah .csv yang kemudian disimpan di *software basisdata* MySQL
4. Perangkat lunak yang digunakan untuk mendukung pengerjaan Tugas Akhir ini adalah bahasa pemrograman *Java*

1.4 Tujuan

Berdasarkan permasalahan yang telah dirumuskan sebelumnya, tujuan penelitian Tugas Akhir ini adalah untuk pengklasteran topik Tugas Akhir mahasiswa ITS berdasarkan abstrak dengan *Rapid Automatic Keyphrase Extraction* (RAKE) dan *Average Linkage Hierarchical Clustering.*

1.5 Manfaat

Manfaat dari penelitian Tugas Akhir ini adalah :

1. Memberikan informasi mengenai *cluster* – *cluster* yang memuat Tugas Akhir berdasarkan kemiripan kata penting
2. Memberikan informasi tambahan kepada mahasiswa ITS mengenai topik - topik yang ada di koleksi dokumen Tugas Akhir di ITS.

1.6 Sistematika Penulisan Tugas Akhir

Sistematika dari penulisan Tugas Akhir ini adalah sebagai berikut :

1. BAB I PENDAHULUAN
   Bab ini menjelaskan tentang gambaran umum dari penulisan Tugas Akhir ini yang meliputi latar belakang masalah, perumusan masalah, batasan masalah, tujuan, manfaat penelitian, dan sistematika penulisan.
2. BAB II TINJAUAN PUSTAKA
   Bab ini berisi tentang materi-materi yang mendukung Tugas Akhir ini, antara lain penelitian terdahulu, *Text mining*, Metode *Rapid Automatic Keyphrase Extraction* (RAKE), *Term Frequncy-Inverse Document Frequency* (TF-IDF), *Clustering* dan *Silhoutte coeffisient.*
3. BAB III METODE PENELITIAN
   Pada bab ini dibahas tentang langkah – langkah dan metode yang digunakan untuk menyelesaikan Tugas Akhir ini.
4. BAB IV PERANCANGAN DAN IMPLEMENTASI PERANGKAT LUNAK
   Pada bab ini akan menguraikan bagaimana tahapan tahapan dalam perancangan implementasi.
5. BAB V HASIL DAN PEMBAHASAN
   Bab ini menjelaskan mengenai hasil pengujian *Rapid Automatic Keyphrase Extraction* (RAKE) untuk ekstraksi kata penting dan *Average Linkage*

Hierarchical Clustering untuk pengklasteran data Tugas Akhir Mahasiswa ITS. Setelah itu dilakukan analisis terhadap hasil implementasi.

6. BAB VI PENUTUP
   Bab ini berisi kesimpulan yang diperoleh dari pembahasan masalah sebelumnya serta saran yang diberikan untuk pengembangan selanjutnya.

BAB II
TINJAUAN PUSTAKA

Pada bab ini dibahas mengenai dasar teori yang digunakan dalam penyusunan Tugas Akhir ini. Dasar teori yang dijelaskan dibagi menjadi beberapa subbab yaitu penelitian terdahulu, *Text mining*, Metode *Rapid Automatic Keyphrase Extraction* (RAKE), *Term Frequncy-Inverse Document Frequency* (TF-IDF), *Clustering, Silhoutte Coefficient*

2.1 Penelitian Terdahulu

Pada penelitian sebelumnya, Nurul Arifin Subandi [4] telah melakukan penelitian tentang *clustering* dokumen skripsi berdasarkan abstrak. Data yang digunakan yaitu skripsi Ilmu Komputer IPB yang terdiri atas 78 dokumen abstrak berbahasa Indonesia dan 113 dokumen abstrak berbahasa Inggris dengan format PDF. Penelitian tersebut menggunakan metode *Bisecting K-Means.* Hasil penelitian menunjukkan bahwa dengan menggunakan *Bisecting K-Means*, nilai *treshold i* (jarak internal *cluster*) terbaik yang dihasilkan untuk *clustering* abstrak bahasa Indonesia adalah 0,67 dengan *rand index* sebesar 0,867 dan nilai *treshold i* terbaik untuk *clustering* abstrak bahasa Inggris adalah 0,55 dengan *rand index* sebesar 0,862. Namun proses *preprocessing* yang digunakan pada penelitian Nurul Arifin Subandi masih menggunakan *lowercase* dan *stopword*, sehingga dibutuhkan waktu yang lama untuk proses *preprocessing*.

Penelitian tentang kemiripan Tugas Akhir berdasarkan abstrak juga telah dilakukan oleh Rosyid pada tahun 2009 [3]. Penelitian tersebut merupakan sistem yang dapat mengetahui kedekatan atau kemiripan judul – judul proyek akhir yang akan diajukan dengan memasukkan judul dan abstrak yang sudah dipilih dan membandingkannya pada proyek akhir teknik informatika yang sudah ada dari tahun 2006 sampai dengan

2009. Namun pada penelitian ini data hanya terbatas pada proyek akhir Departemen Informatika. Selain itu proses *text mining* yang digunakan hanya tahapan *tokenizing* dan *filtering* yang mengharuskan mengolah semua data sehingga membutuhkan waktu komputasi yang lama. Selanjutkan dilakukan proses *clustering* dengan menggunakan metode *Single Linkage Hierarchical* untuk membentuk 9 *cluster* bidang teknik informatika, setelah terbentuknya 9 *cluster* tersebut maka akan dilakukkan proses *inner product*, yaitu perkalian tiap *cluster* yang sudah terbentuk tersebut dengan input berupa judul dan abstrak dari pengajuan judul proyek akhir yang telah melalui proses *text mining*. Dari empat puluh kali percobaan dengan inputan yang berbeda di setiap percobaan hasilnya memberikan kesimpulan bahwa pada umunya penentuan kemiripan topik proyek akhir berdasarkan abstrak pada jurusan teknik informatika dengan metode *single linkage hierarchical* dapat digunakan untuk mengetahui kemiripan atau kedekatan judul proyek akhir sesuai dengan inputan. Semakin atas urutan/ranking dari output judul yang dihasilkan maka semakin mendekati dengan inputan abstrak yang diinputkan.

Penelitian lain telah dilakukan oleh Tahta Alfina, Budi Santoso dan Ali Ridho Barakbah pada tahun 2012 [5]. Peneliti menganalisa perbandingan metode *Hierarchical Clustering, K-Means* dan gabungan keduanya dalam *cluster* data. Data yang digunakan dalam penelitian ini adalah data teks yaitu data problem kerja praktek Jurusan Teknik Industri ITS yang disampaikan oleh mahasiswanya melalui forum diskusi jejaring sosial *facebook*. Akan tetapi pada penelitian tersebut digunakan algoritma *document clustering* sederhana. *Keyword* yang digunakan ditentukan secara manual oleh peneliti sehingga domain teks yang akan dibawa kedalam suatu *cluster* bersifat spesifik. Padahal *text mining* digunakan untuk mengelompokkan data dimana domainnya bersifat bebas. Hasil penelitan ini yaitu pengujian yang dilakukan menggunakan

koefisien korelasi *cophenetic* menghasilkan metode *clustering* terbaik adalah metode *average linkage hierarchical clustering*.

Penelitian lain mengenai *clustering* dengan menggunakan *average linkage hierarchical clustering* dilakukan oleh Sofya Laeli [6]. Penelitian yang dilakukan pada tahun 2014 ini berjudul Analisis *Cluster* dengan *Average Linkage Method* dan *Ward's Method* untuk data responden nasabah Asuransi Jiwa Unit Link. Penelitian ini bertujuan untuk mengetahui langkah-langkah analisis *cluster* dengan metode *average linkage* dan metode *Ward*, serta membandingkan hasil analisis kedua metode tersebut untuk meng*cluster*kan beberapa responden terkait alasan dalam memutuskan untuk membeli produk Asuransi Jiwa Unit Link. Hasil penelitian ini menunjukkan bahwa metode *average linkage* memiliki kinerja lebih baik daripada metode *Ward*. Namun banyak data yang digunakan sudah ditentukan berdasarkan banyaknya jumlah variabel yeng diteliti. Variabel – variabel tersebut merepresentasikan jawaban dari kuisioner/angket yang diajukan ke responden sehingga jumlah *cluster* awal sudah ditentukan sebelumnya. Jika data yang digunakan lebih beragam dengan jumlah *cluster* awal yang lebih banyak maka hasil yang didapatkan belum tentu sama.

2.2 *Text Mining*

Text mining dapat diartikan sebagai penemuan informasi yang baru dan tidak diketahui sebelumnya oleh komputer, secara otomatis mengekstrak informasi dari sumber – sumber yang berbeda. Kunci dari proses ini adalah menggabungkan informasi yang berhasil diekstraksi dari berbagai sumber [7]. Tujuan utama *text mining* adalah mendukung proses *knowledge discovery* pada koleksi dokumen yang besar. *Teks mining* dapat dipandang sebagai suatu perluasan dari *data mining* atau *knowledge-discovery in database* (KDD), yang mencoba untuk menemukan pola-pola menarik dari basis data berskala besar. Namun *text mining* memiliki potensi *komersil* yang lebih tinggi

dibandingkan dengan *data mining*, karena kebanyakan format alami dari penyimpanan informasi adalah berupa teks. *Text mining* menggunakan informasi teks tak terstruktur [8].

Perbedaan mendasar antara *text mining* dan *data mining* terletak pada sumber data yang digunakan. Pada *data mining*, pola-pola diekstrak dari basis data yang terstruktur, sedangkan di *text mining*, pola-pola diekstrak dari data tekstual (*natural language*). Secara umum, basis data didesain untuk program dengan tujuan melakukan pemrosesan secara otomatis, sedangkan teks ditulis untuk dibaca langsung oleh manusia.

2.3 Metode *Rapid Automatic Keyphrase Extraction* (RAKE)

Metode *Rapid Automatic Keyphrase Extraction* (RAKE) merupakan metode yang *unsupervised*, serta tidak tergantung pada bahasa. Metode *Rapid Automatic Keyphrase Extraction* (RAKE) adalah metode yang menggunakan pendekatan berbasis dokumen individu yang mampu mengelompokkan topik penelitian tanpa bergantung pada koleksi dokumen lain [9]. Metode RAKE memperhatikan asosiasi kata dengan menghitung matriks kemunculan bersama satu dengan yang lain. Matriks tersebut digunakan untuk mengukur skor kandidat kata penting untuk kemudian dilakukan perengkingan [10]. Kata penting adalah bagian dari kalimat yang merepresentasikan ide utama dari sebuah dokumen. Kata penting dimaksudkan untuk sebuah kata atau lebih sebagai kunci, sedangkan frase penting adalah dua kata atau lebih sebagai kunci. Metode RAKE dikembangkan pada pengamatan bahwa kata penting sering kali terdiri dari beberapa kata tetapi jarang terdiri dari *stopword* seperti dan, itu, ini, dll. *Stopword* biasanya dihapus dalam sistem pengambalian informasi karena dianggap tidak informatif atau kurang bermakna [11].

Metode RAKE memiliki tahapan sebagai berikut [12] :

1. Ekstraksi kandidat
   Ekstraksi kandidat kata penting dimulai dengan memisahkan teks menggunakan *stopword* dan tanda baca.
2. Menghitung matriks *co-ocurrence*
   Setelah kandidat kata penting didapatkan, langkah selanjutnya adalah menghitung matriks *co-ocurrence*. Matriks *co-ocurrence* memetakan frekuensi kemunculan suatu kata dan frase kata penting. Berikut cuplikan dari matriks *co-ocurrence* :

	cell	Dssc	dye	ekstraksi	sensitized	...
cell	2		2			...
dssc		8				...
dye	2		6			...
ekstraksi				1		...
sensitized	2		2		2	...
...	...	...	...	...	...	...

3. Menghitung rasio
   Nilai rasio merupakan perbandingan antara derajat kata dengan frekuensi kata. Frekuensi kata adalah jumlah kemunculan kata dalam dokumen atau dapat diambil dari skor diagonal kata pada matriks *co-ocurrence*. Derajat kata adalah jumlah kemunculan kata tersebut pada dokumen ditambah jumlah frase yang mengandung kata tersebut. Derajat kata pada matriks *co-ocurrence* didapat dari penjumlahan skor kata pada satu kolom atau satu baris. Misal kata *algorithms* muncul sendiri sekali dan juga muncul frase : *corresponding algorithms*, maka kata *algorithms* memiliki derajat kata sebanyak 2+1 = 3. Hal ini karena kata *algorithms* muncul 2 kali yakni 1 kali dari *algorithms* dan 1 kali dalam *corresponding*

algorithms dan mucul dalam frase *corresponding algorithms* sebanyak 1 kali. Rasio dari kata w, dapat dihitung dengan menggunakan persamaan berikut ini.

$$rasio\ (w) = \frac{deg(w)}{fre(w)}\ , \qquad (2.1)$$

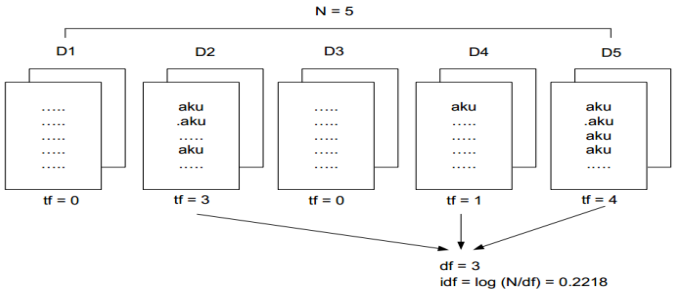
dengan:
$deg(w)$: derajat kata
$fre(w)$: frekuensi kata
w : kata

4. Menghitung nilai fitur dasar
   Setelah menghitung rasio kata proses selanjutnya, menghitung nilai fitur dasar yaitu dengan cara masing – masing kandidat diberi skor dari hasil penjumlahan skor rasio kata yang dimiliki. Misalkan terdapat frase *corresponding algorithms*, kata *corresponding* memiliki nilai rasio 2 dan kata *algorithms* memiliki nilai rasio 2,5. Sehingga nilai skor akhir frase *corresponding algorithms* adalah 2 + 2,5 = 4,5. Setelah pemberian skor pada kandidat, dilakukan pengurutan berdasarkan skor akhir dari tertinggi sampai terendah.
5. Pemilihan kandidat frase penting dengan skor tertinggi
   Setelah kandidat kata/frase penting diberi skor, selanjutnya dipilih sejumlah k kandidat dengan skor tertinggi.

RAKE cenderung mendukung frase yang panjang karena semakin panjang frase, semakin tinggi skor yang dimiliki. Oleh karena itu, penting untuk memilih *stopword* dengan hati-hati. Hal ini bertujuan untuk menghindari frase yang panjang dengan relevansi yang kurang untuk dipilih sebagai kata penting [11].

2.4 *Term Frequency-Inverse Document Frequency* (TF-IDF)

Model ruang vektor untuk koleksi dokumen mengandaikan dokumen *d* sebagai sebuah vektor dalam *term space*. Dalam rangka membangun model vektor, perlu dilakukan proses pembobotan *term* (kata penting). Skema pembobotan yang paling banyak digunakan adalah skema *term frequency-inverse document frequency* (TF-IDF). Pembobotan *term* (*term Weighting*) bertujuan untuk menentukan bobot setiap *term*. Perhitungan bobot *term* memerlukan dua hal yaitu *Term Frequency* (tf) dan *Inverse Document Frequency* (idf). *Term Frequency* (tf) merupakan frekuensi kemunculan suatu kata (*term*) dalam suatu dokumen. Nilai tf bervariasi ditiap dokumen bergantung pada kemunculan kata di suatu dokumen. Besar nilai tf sebanding dengan tingkat kemunculan *term* di dokumen. Semakin sering *term* muncul pada suatu dokumen, semakin besar pula nilai tf pada dokumen tersebut dan semakin jarang *term* muncul semakin kecil pula nilai tf. Selain *Term Frequency* diperlukan pula *Inverse Document Frequency* (idf) pada pembobotan *term*. *Inverse Document Frequency* (idf) merupakan frekuensi kemunculan *term* pada keseluruhan dokumen. Nilai idf berkaitan dengan distribusi *term* di berbagai dokumen. Ilustrasi tf-idf ditunjukkan pada Gambar 2.1 [13]. Pada Gambar 2.1 menjelaskan bahwa terdapat 5 dokumen yaitu D1, D2, D3, D4 dan D5 dengan *term* (kata penting) yang dihitung adalah aku. Pada D1 tidak mengandung kata penting aku sama sekali sehingga nilai tf = 0, sementara pada D2 mengandung kata penting aku sebanyak 3 sehingga tf pada dokumen D2 adalah 3, pada D3 tidak mengandung kata penting aku sama sekali sehingga nilai tf = 0, pada D4 mengandung kata penting aku satu kali sehingga nilai tf = 1, dan pada D5 mengandung kata penting aku sebanyak 4 kali sehingga nilai tf = 4. Pada Gambar 2.1 menunjukkan bahwa dokumen yang mengandung kata penting aku adalah 3 dokumen sehingga niai df adalah 3.



Gambar 2.1 Ilustrasi tf-idf

Menghitung nilai idf [13] :

$$idf = \log \frac{N}{df} \ , \tag{2.2}$$

dengan :

D1, …, D5	: dokumen
tf	: banyaknya *term* (kata penting) yang dicari pada setiap dokumen
N	: total dokumen
df	: banyaknya dokumen yang mengandung *term* (kata penting) yang dicari

Persamaan menghitung nilai tf-idf adalah [13] :

$$W_{i,j} = tf_{i,j} \times idf_i = tf_{i,j} \times \log\left(\frac{N}{df_i}\right) \ , \tag{2.3}$$

dengan :

$W_{i,j}$: bobot *term* (kata penting) ke-i terhadap dokumen ke-j

$tf_{i,j}$: jumlah kemunculan *term* i (kata penting) di dokumen j

N : jumlah dokumen secara keseluruhan

df_i : jumlah dokumen yang mengandung *term* (kata penting) i
idf_i : *Inverse Document Frequency* yang mengandung *term* (kata penting) i

Perhitungan bobot dari *term* tertentu dalam sebuah dokumen dengan menggunakan tf x idf menunjukkan bahwa deskripsi terbaik dari dokumen adalah *term* yang banyak muncul dalam dokumen tersebut dan sangat sedikit muncul pada dokumen lain [14].

2.5 *Clustering*

Clustering adalah salah satu teknik *data mining* yang bertujuan untuk mengidentifikasi sekelompok obyek yang mempunyai kemiripan karakteristik tertentu yang dapat dipisahkan dengan kelompok obyek lainnya, sehingga obyek yang berada dalam kelompok yang sama relatif lebih homogen daripada objek yang berada pada kelompok yang berbeda [15].

Ada beberapa pendekatan yang digunakan dalam mengembangkan metode *clustering*. Dua pendekatan utama adalah *clustering* dengan pendekatan *partisi* dan *hirarki*. *Clustering* dengan pendekatan *partisi* atau sering disebut dengan *partition-based clustering* mengelompokkan data dengan memilah-milah data yang dianalisa ke dalam *cluster-cluster* yang ada. *Clustering* dengan pendekatan *hirarki* atau sering disebut dengan *hierarchical clustering* mengelompokkan data dengan membuat suatu *hirarki* berupa diagram dimana data yang mirip akan ditempatkan pada *hirarki* yang berdekatan dan yang tidak pada *hirarki* yang berjauhan. Di samping kedua pendekatan tersebut, ada juga *clustering* dengan pendekatan *automatic mapping* (*Self-Organising Map*/SOM) [15].

2.5.1 Metode *Hierarchical Clustering*

Metode hierarki (*hierarchical method)* adalah suatu metode pada analisis *cluster* yang membentuk tingkatan

tertentu seperti pada struktur pohon karena proses pengklasterannya dilakukan secara bertingkat/bertahap. Hasil pengklasteran dengan metode hirarki dapat disajikan dalam bentuk dendogram. Dendogram adalah representasi visual dari langkah langkah dalam analisis *cluster* yang menunjukkan bagaimana *cluster* terbentuk dan nilai koefisien jarak pada setiap langkah. Angka disebelah kanan adalah obyek penelitian, dimana obyek- obyek tersebut dihubungkan oleh garis dengan obyek yang lain sehingga pada akhirnya akan membentuk satu *cluster* [6].

Metode - metode yang bisa digunakan dalam metode hirarki adalah metode agglomeratif (*agglomerative method)* dan metode devisif (*devisive method)*.

a. Metode Agglomeratif

Algoritma umum *Hierarchical Agglomerative Clustering* dimulai dengan setiap item dianggap satu *cluster* tersendiri dan secara iteratif menggabungkan *cluster – cluster* sampai semua item berada dalam satu *cluster*. Perbedaan algoritma *Hierarchical Agglomerative Clustering* terdapat dalam bagaimana *cluster* digabungkan pada tiap - tiap tingkat[16]. Metode agglomeratif sendiri masih ada beberapa macam yaitu metode *Single Linkage*, metode *Complete Linkage*, metode *Centroid Linkage*, metode *Avarage Linkage* [6].

Tahap-tahap peng*cluster*an data dengan menggunakan metode *agglomerative hierarchical clustering* [16]:

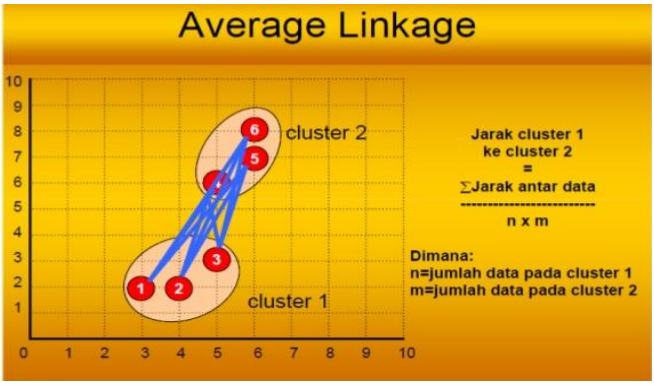
1. Dimulai dengan menetapkan tiap-tiap data menjadi sebuah *cluster*, sehingga jika ada N = jumlah data, berarti terdapat N *cluster.*
2. Hitung jarak (*similarity*) antar *cluster.*
3. Cari pasangan *cluster* terdekat dan gabungkan sehingga menjadi satu *cluster* baru
4. Hitung jarak (*similarity*) antar *cluster* baru dengan tiap-tiap *cluster* yang lama. Hitung jarak menggunakan metode *agglomerative hierarchical clustering* yang telah ditentukan.

5. Ulangi langkah 3 dan 4 sampai semua data berada dalam sebuah *cluster* tunggal berukuran N atau proses dapat pula berhenti jika telah mencapai batasan kondisi tertentu.

b. Metode Devisif
Proses dalam metode divisif berkebalikan dengan metode agglomeratif. Metode ini dimulai dengan satu *cluster* besar yang mencakup semua obyek pengamatan. Selanjutnya, secara bertahap obyek yang mempunyai ketidakmiripan cukup besar akan dipisahkan ke dalam *cluster-cluster* yang berbeda. Proses dilakukan sehingga terbentuk sejumlah *cluster* yang diinginkan, seperti, dua *cluster*, tiga *cluster*, dan seterusnya [6].

2.5.1.1 Metode *Average Linkage Hierarchical Clustering*

Average Linkage adalah proses *clustering* yang didasarkan pada jarak rata – rata antar obyeknya [3]. Prosedur ini hampir sama dengan *single linkage* maupun *complete linkage*, namun kriteria yang digunakan adalah rata – rata jarak seluruh individu dalam suatu *cluster* dengan jarak seluruh individu dalam *cluster* yang lain [6]. Metode ini relatif yang terbaik dari metode – metode *hierarchical*. Namun, ini harus dibayar dengan waktu komputasi yang paling tinggi dibandingkan dengan metode – metode *hierarchical* yang lain. Ilustrasi dari *Average Linkage Hierarchical Clustering* digambarkan seperti Gambar 2.2 [3]:



Gambar 2. 2 Ilustrasi *Average Linkage Hierarchical Clustering*

Berdasarkan tahap pengklasteran menggunakan *agglomerative hierarchical clustering*, maka tahap pengklasteran menggunakan *average linkage hierarchical* sebagai berikut :

1. Dimulai dengan menetapkan setiap dokumen sebagai *cluster*. Jika n = jumlah dokumen, c = jumlah *cluster*, berarti c = n.
2. Menghitung jarak (*similarity*) antar *cluster*
   Pada penelitian ini jarak (*similarity*) yang digunakan adalah *cosine similarity*. Berikut persamaan dari *cosine similarity* [17] :

$$sim(d_p, d_q) = \frac{d_p \cdot d_q}{\|d_p\|\|d_q\|} = \frac{\sum_{k=1}^{n} w_{p,k}\, w_{q,k}}{\sqrt{\sum_{k=1}^{n} w_{p,k}^2}\sqrt{\sum_{k=1}^{n} w_{q,k}^2}}, \qquad (2.4)$$

dengan :

d_p : vektor dokumen ke p

d_q : vektor dokumen ke q

n : banyaknya kata penting
$w_{p,k}$: bobot kata penting ke k pada dokumen p
$w_{q,k}$: bobot kata penting ke k pada dokumen q
$\|d_p\|$: hasil kali dalam dari vektor dokumen ke p
$\|d_q\|$: hasil kali dalam dari vektor dokumen ke q

3. Cari 2 *cluster* terdekat (paling mirip) yaitu dengan mencari *similarity* terbesar dan gabungkan sehingga menjadi satu *cluster* baru.
4. Hitung jarak (*similarity*) rata - rata antar *cluster* baru dengan tiap-tiap *cluster* yang lama dengan menggunakan *average linkage hierarchical clustering* dengan persamaan berikut [18]:

$$dist_{avg}\ (C_r, C_s) = \frac{1}{n_r n_s} \sum_{o \in C_r,\ o' \in C_s} |o - o'|\ , \qquad (2.5)$$

dengan :
$dist_{avg}\ (C_r, C_s)$: jarak rata-rata antar *cluster*
C_r : *cluster* C_r
C_s : *cluster* C_s
$|\ o - o'|$: jarak antara dua dokumen
n_r : banyaknya dokumen pada *cluster* C_r
n_B : banyaknya dokumen pada *cluster* C_s

Jarak antara dua dokumen dihitung menggunakan Persamaan 2.4 sehingga didapat persamaan :

$$dist_{avg}(C_r, C_s) = \frac{1}{n_r n_s} \sum_{p \in C_r} \sum_{q \in C_s} sim(d_p, d_q)\ , \quad (2.6)$$

dengan $sim(\ d_p, d_q\ )$ merupakan jarak antara obyek d_p pada *cluster* C_r dan obyek d_q pada *cluster* C_s, dimana $p \in C_r$ dan $q \in C_s$

5. Ulangi langkah 3 dan 4 sampai semua data berada dalam sebuah *cluster* tunggal berukuran N atau proses dapat pula berhenti jika telah mencapai batasan kondisi tertentu.

2.6 *Silhoutte Coefficient*

Silhouette Coefficient adalah sebuah teknik yang digunakan untuk mengukur seberapa baik letak objek dalam *cluster*. Metode ini merupakan gabungan dari metode *cohesion* dan *separation*. Berikut ini adalah tahapan perhitungan rumus *Silhouette Coefficient*:

1. Hitung rata-rata jarak dari suatu dokumen misalkan *m* dengan semua dokumen lain yang berada dalam satu *cluster* [19]

$$a(m) = \frac{1}{|A|}\sum_{\substack{m,n\in A \\ m\neq n}} d(m,n) \ , \tag{2.7}$$

dengan :

$a(m)$: rata - rata jarak suatu dokumen dengan semua dokumen lain yang berada dalam satu *cluster* A

$d(m,n)$: jarak antar dokumen *m* dengan *n* dalam *cluster* A

n : dokumen lain, selain m dalam satu *cluster* A

$|A|$: banyaknya dokumen dalam *cluster* A

2. Hitung rata-rata jarak dari dokumen *m* tersebut dengan semua dokumen di *cluster* lain, dan diambil nilai terkecilnya [18]

$$d(m,B) = \frac{1}{|B|}\sum_{\substack{l\in B \\ m\in A}} d(m,l) \ , \tag{2.8}$$

$$b(m) = \min d(m,B), \ \ B \neq A \ , \tag{2.9}$$

dengan:

$d(m, B)$: jarak rata-rata antar dokumen *m* dengan semua objek pada *cluster* lain (*cluster* B), B ≠ A

$d(m, l)$: jarak antar dokumen *m* dengan *l*

n : dokumen lain dalam *cluster* lain, selain *cluster* A

$b(m)$: nilai terkecil dari $d(m, B)$

$|B|$: banyaknya dokumen dalam *cluster* B

3. Nilai *Silhouette Coefficient* dihitung menggunakan persamaan berikut ini [18] :

$$s(m) = \frac{b(m)-a(m)}{\max\{a(m),b(m)\}} , \qquad (2.10)$$

$s(m)$: *Silhoutte Coefficient*

Rata-rata $s(m)$ dari seluruh data dalam suatu *cluster* menunjukkan seberapa dekat kemiripan data dalam suatu *cluster* yang juga menunjukkan seberapa tepat data telah dikelompokkan. Nilai *silhoutte coefficient* adalah antara -1 dan 1. Jika nilai *silhoutte coefficient* semakin positif menunjukkan semakin baik atau semakin tepat data dikelompokkan, namun jika semakin negatif menunjukkan kurang tepat data dikelompokkan [19].

BAB III
METODE PENELITIAN

Pada bab ini dijelaskan langkah-langkah yang digunakan dalam penyusunan Tugas Akhir. Disamping itu, dijelaskan pula prosedur dan proses pelaksanaan tiap-tiap langkah yang dilakukan dalam menyelesaikan Tugas Akhir.

3.1 Studi Literatur

Pada tahap ini dilakukan pengumpulan informasi mengenai beberapa hal berikut :

1. Pengumpulan informasi mengenai cara ekstraksi kata penting suatu dokumen menggunakan metode *Rapid Automatic Keyphrase Extraction* (*RAKE*)
2. Pengumpulan informasi mengenai cara pembentukan vektor konsep dengan *kata penting frequency-invers dokument frequency* (TF-IDF)
3. Pengumpulan informasi mengenai metode *clustering* menggunakan *Average Linkage Hierarchical Clustering* yang digunakan untuk mengelompokkan konsep – konsep yang mirip menjadi satu *cluster*

3.2 Pengumpulan Data

Pada tahap ini dilakukan pengumpulan data dari *database* Tugas Akhir mahasiswa ITS pada tahun 2012 sampai dengan 2016 sebanyak 6.916 data. Atribut yang yang digunakan yaitu abstrak, NRP mahasiswa, jurusan dan tahun terbit.

3.3 Ekstraksi Kata Penting

Pada tahap ini akan dilakukan ekstraksi kata penting menggunakan metode *Rapid Automatic Keyphrase* (RAKE). Terdapat lima tahap utama yaitu ektraksi kandidat kata penting, menghitung matriks *co-ocurrence*, menghitung nilai rasio, menghitung nilai fitur dasar, memilih frase penting dengan

nilai fitur tertinggi. Terdapat pembuatan perangkat lunak untuk mendukung proses ekstraksi kata penting.

3.4 Pembentukan Vektor Konsep

Pada tahap ini akan dilakukan pembentukan vektor konsep dengan melakukan pembobotan dengan *Term Frequency-Invers Dokument Frequency* (TF-IDF). Terdapat dua tahapan utama yaitu membangun matriks kemunculan tiap kata penting pada tiap dokumen (*tf*) dan menghitung nilai *Invers Document Frequency (idf)*. Terdapat pembuatan perangkat lunak untuk mendukung proses pembentukan vektor konsep.

3.5 Pengklasteran

Pada tahap ini akan dilakukan pengklasteran yang sudah didapatkan dari tahap sebelumnya dan kemudian menentukan topik pada *cluster*. Metode pengklasteran yang digunakan pada penelitian ini adalah metode *Average Linkage Hierarchical Clustering.*

3.6 Analisa Hasil dan Pembahasan

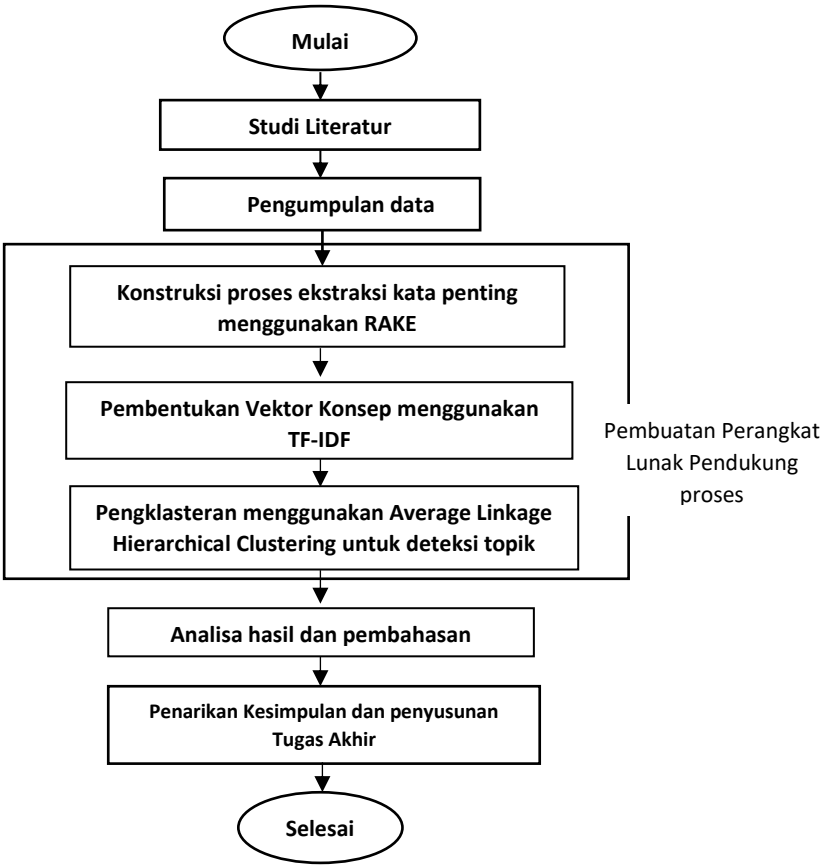
Hasil dari proses *clustering* berupa informasi mengenai topik - topik yang ada pada *cluster* dan akan dilakukan evaluasi untuk validasi *clustering* dengan menggunakan *Silhoutte Coefficient*

3.7 Penarikan Kesimpulan dan Penyusunan Laporan Tugas Akhir

Dalam tahap akhir penelitian ini, dilakukan penarikan kesimpulan dan penyusunan laporan Tugas Akhir dari hasil analisis dan pembahasan yang telah dilakukan mengenai pengklasteran laporan Tugas Akhir berdasarkan abstrak meggunakan metode *Rapid Automatic Keyphrase Extraction* dan *Average Linkage Hierarchical Clustering.*

Pembuatan perangkat lunak dalam tahap ekstraksi kata penting, pembentukan vektor konsep yang dilanjutkan proses pengklasteran laporan Tugas Akhir berdasarkan abstrak adalah perangkat lunak sebagai *tools* pendukung pengerjaan tugas akhir ini.

Berikut adalah alur pengerjaan Tugas Akhir ini yang ditunjukkan pada Gambar 3.1:



Gambar 3.1 Alur pengerjaan Tugas Akhir

BAB IV
PERANCANGAN DAN IMPLEMENTASI PERANGKAT LUNAK

Bab ini menjelaskan rancangan yang digunakan sebagai acuan untuk implementasi sistem. Perancangan implementasi menggambarkan proses rancang bangun secara terperinci dari awal tahap pengumpulan data hingga proses *clustering* mengunakan metode *Average Linkage Hierarchical Clustering*

4.1 Perancangan Data

Tahap ini bertujuan untuk menjelaskan data – data yang digunakan dalam program. Data yang diperoleh penulis merupakan data Tugas Akhir yang ada di ITS pada tahun 2012 – 2016. Terdapat dua jenis data yang digunakan dalam perangkat lunak ini yaitu data masukan dan data keluaran.

4.1.1 Data Masukan

Data masukan adalah data – data yang digunakan sebagai inputan/masukan ke program. Inputan atau masukan ini yang kemudian akan diolah oleh aplikasi melalui tahap tahap tertentu sehingga menghasilkan keluaran yang diinginkan. Data masukan yang digunakan yaitu :

a. Data Tugas Akhir mahasiswa ITS yang telah disimpan di *database*. Terdapat 6.916 data yang disimpan. *Database* Tugas Akhir mahasiswa ITS memiliki beberapa atribut yaitu :
    1. NRP merupakan id dokumen
    2. abstrak merupakan abstrak
    3. tahun merupakan tahun rilis buku Tugas Akhir
    4. jurusan merupakan jurusan
b. Data kumpulan *stopword* yang telah tersimpan di dalam *database*. Terdapat 1.135 data *stopword*.

4.1.2 Data Keluaran

Data keluaran merupakan data yang dihasilkan oleh aplikasi setelah proses – proses tertentu selesai dilakukan. Terdapat beberapa data keluaran pada aplikasi ini yaitu :

a. Data hasil ekstraksi konsep menggunakan *Term Frequency-Invers Dokument Frequency* (TF-IDF)
b. Data hasil *clustering* dengan menggunakan metode *Average Linkage Hierarchical Clustering*
c. Data hasil perhitungan akurasi terhadap *clustering* menggunakan metode *Silhoutte Coefficient*

4.2 Peralatan

Peralatan utama yang digunakan untuk menyelesaikan penelitian ini berupa perangkat keras dan perangkat lunak yaitu sebagai berikut :

1. Perangkat keras berupa *Personal Computer* (PC) dengan spesifikasi :
   - Processor Intel Core i5-6200U 2,8GHz
   - RAM 4GB
2. Perangkat Lunak yang digunakan adalah
   - Java Netbeans IDE 7.4
   - XAMPP versi 3.2.2

4.3 Perancangan Proses

Terdapat tiga proses yang dapat dilakukan oleh pengguna dalam perangkat lunak ini. Proses – proses tersebut antara lain :

4.3.1 Ekstraksi Kata Penting

Proses ekstraksi kata penting menggunakan metode *Rapid Automatic Keyphrase Extraction* (RAKE) memiliki 5 tahapan utama yaitu ektraksi kandidat kata penting, menghitung matriks *co-ocurrence*, menghitung nilai rasio, menghitung nilai fitur dasar, memilih kata penting dengan nilai fitur tertinggi.

Langkah langkah ektraksi kata penting sebagai berikut :
Input : Abstrak
Output : Kata penting
Proses Ekstraksi Kata Penting :

1. Pengguna memasukan jumlah kata penting yang digunakan.
2. Baca abstrak dokumen *project* dari *database*
3. Ubah seluruh huruf pada abstrak menjadi huruf kecil
4. Hilangkan seluruh karakter angka
5. Pisahkan isi abstrak menurut tanda baca
6. Simpan seluruh kata atau frase yang memiliki panjang lebih dari 2 huruf sebagai representasi dokumen
7. Untuk setiap kata atau frase yang telah disimpan pisahkan menurut *stopword*
8. Simpan seluruh kata dan frase sebagai kandidat kata penting
9. Hitung frekuensi kemunculan setiap kata di dalam dokumen $freq(w)$
10. Hitung nilai *degree* setiap kata $\deg(w)$
11. Hitung nilai rasio kata, $rasio(w) = \frac{\deg(w)}{freq(w)}$ setiap kata
12. Hitung nilai fitur setiap kandidat kata penting dengan cara menambahkan nilai rasio tiap kata yang ada pada kandidat kata penting
13. Urutkan nilai fitur kandidat kata penting dari kecil ke besar
14. Simpan kata penting sebanyak input jumlah kata penting yang dimasukkan oleh pengguna ke dalam *database*.
15. Ulangi untuk setiap dokumen di dalam *database*

Berikut merupakan contoh ektraksi kata penting menggunakan *Rapid Automatic Keyphrase Extraction* (RAKE) :

Tabel 4.1 Contoh abstrak yang belum diolah

Telah dilakukan fabrikasi dan karakterisasi Dye Sensitized Solar Cell (DSSC) dengan menggunakan ekstraksi daging buah naga merah sebagai dye sensitizer. DSSC merupakan sel surya berbasis fotoelektrokimia dimana digunakan zat warna organik sebagai penyerap cahaya matahari dan semikonduktor anorganik sebagai tempat terjadinya separasi muatan listrik. DSSC dapat mengkonversi cahaya matahari menjadi energi listrik dengan menggunakan elektrolit sebagai transfer muatan. Penelitian dilakukan dengan variasi dye dan variasi elektrolit pada ketinggian 5 cm dan 10 cm. Dilakukan karakterisasi pengukuran tegangan dan arus terhadap waktu dengan sumber cahaya halogen 6 volt 30 watt. Dari hasil pengujian diperoleh tegangan dan arus yang lebih tinggi dan stabil pada DSSC dengan dye(100 gr daging buah naga merah+5 ml aquades) dari pada DSSC dengan dye(100 gr daging buah naga merah+10 ml aquades). Sedangkan pada variasi elektrolit, tegangan dan arus yang dihasilkan oleh DSSC dengan elektrolit(6 gr KI+3 ml iodin solution 10%) lebih tinggi dan stabil dari pada DSSC dengan elektrolit (3 gr KI+3 ml iodin solution 10% dan 3 gr KI+6 ml iodin solution 10%). Sel Surya; Dye Sensitized Solar Cell (DSSC); Buah Naga Merah (Hylocereus Polyrhizus)

Tabel 4.2 Contoh hasil abstrak yang telah dirubah menjadi huruf kecil dan penghilangan karakter angka

telah dilakukan fabrikasi dan karakterisasi dye sensitized solar cell (dssc) dengan menggunakan ekstraksi daging buah naga merah sebagai dye sensitizer. dssc merupakan sel surya berbasis fotoelektrokimia dimana digunakan zat warna

organik sebagai penyerap cahaya matahari dan semikonduktor anorganik sebagai tempat terjadinya separasi muatan listrik. dssc dapat mengkonversi cahaya matahari menjadi energi listrik dengan menggunakan elektrolit sebagai transfer muatan. penelitian dilakukan dengan variasi dye dan variasi elektrolit pada ketinggian cm dan cm. dilakukan karakterisasi pengukuran tegangan dan arus terhadap waktu dengan sumber cahaya halogen volt watt. dari hasil pengujian diperoleh tegangan dan arus yang lebih tinggi dan stabil pada dssc dengan dye(gr daging buah naga merah+ ml aquades) dari pada dssc dengan dye(gr daging buah naga merah+ ml aquades). sedangkan pada variasi elektrolit, tegangan dan arus yang dihasilkan oleh dssc dengan elektrolit(gr ki+ ml iodin solution %) lebih tinggi dan stabil dari pada dssc dengan elektrolit (gr ki+ ml iodin solution % dan gr ki+ ml iodin solution %). sel surya; dye sensitized solar cell (dssc); buah naga merah (hylocereus polyrhizus)

Tabel 4.3 Beberapa hasil pemotongan abstrak berdasarkan tanda baca dan penghapusan kata/frase yang memiliki panjang kurang dari 2 karakter

telah dilakukan fabrikasi dan karakterisasi dye sensitized solar cell
Dssc
dengan menggunakan ekstraksi daging buah naga merah sebagai dye sensitizer
dssc merupakan sel surya berbasis fotoelektrokimia dimana digunakan zat warna organik sebagai penyerap cahaya matahari dan semikonduktor anorganik sebagai tempat terjadinya separasi muatan listrik
dssc dapat mengkonversi cahaya matahari menjadi energi listrik dengan menggunakan elektrolit sebagai transfer muatan

penelitian dilakukan dengan variasi dye dan variasi elektrolit pada ketinggian dan
dilakukan karakterisasi pengukuran tegangan dan arus terhadap waktu dengan sumber cahaya halogen volt watt
....

Tabel 4.4 Beberapa hasil pemotongan abstrak berdasarkan *stopword*

Fabrikasi
karakterisasi dye sensitized solar cell
Dssc
Ekstraksi
buah naga merah
dye sensitizer
Dssc
sel surya berbasis fotoelektrokimia
...

Tabel 4.5 Hasil kandidat kata penting RAKE

fabrikasi karakterisasi dye sensitized solar cell dssc ekstraksi buah naga merah dye sensitizer sel surya berbasis fotoelektrokimia zat warna organik penyerap cahaya matahari semikonduktor anorganik separasi muatan listrik mengkonversi cahaya matahari energi listrik elektrolit

transfer muatan
variasi dye
variasi elektrolit
ketinggian
karakterisasi pengukuran tegangan
arus
sumber cahaya halogen volt watt
stabil
dye
aquades
tegangan
iodin solution
sel surya
dye sensitized solar cell
hylocereus polyrhizus

Tabel 4.6 Contoh beberapa kandidat kata penting beserta frekuensi, degree dan rasio

Kandidat Kata Penting	**Frekuensi**	**Degree**	**Rasio**
anorganik	1,0	2,0	2,0
aquades	2,0	2,0	1,0
arus	3,0	3,0	1,0
berbasis	1,0	4,0	4,0
buah	4,0	12,0	3,0
cahaya	3,0	11,0	3,6666
cell	2,0	9,0	4,5
dssc	8,0	8,0	1,0
dye	6,0	15,0	2,5
ekstraksi	1,0	1,0	1,0
elektrolit	5,0	7,0	1,4
energi	1,0	2,0	2,0
fabrikasi	1,0	1,0	1,0
fotoelektrokimia	1,0	4,0	4,0

halogen	1,0	5,0	5,0
hylocereus	1,0	2,0	2,0
iodin	3,0	6,0	2,0
karakterisasi	2,0	8,0	4,0
ketinggian	1,0	1,0	1,0
listrik	2,0	5,0	2,5
matahari	2,0	6,0	3,0
mengkonversi	1,0	3,0	3,0
merah	4,0	12,0	3,0
muatan	2,0	5,0	2,5
naga	4,0	12,0	3,0
...	...	...	...

Tabel 4.7 Contoh beberapa kandidat kata penting beserta skornya

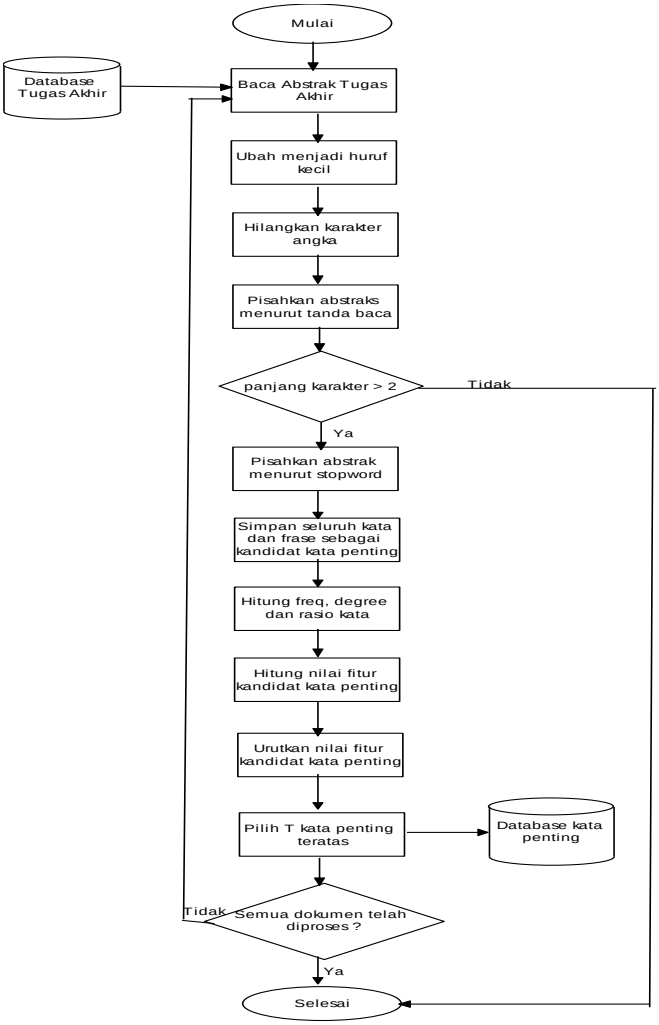
Kandidat Kata Penting	**Skor**
Fabrikasi	1,0
karakterisasi dye sensitized solar cell	18,5
Dssc	1,0
Ekstraksi	1,0
buah naga merah	9,0
dye sensitizer	4,5
sel surya berbasis fotoelektrokimia	14,0
zat warna organik	9,0
penyerap cahaya matahari	9,6666
semikonduktor anorganik	4,0
separasi muatan listrik	8,6666
mengkonversi cahaya matahari	9,6666
energi listrik	4,5
...	...

Tabel 4.8 Contoh kandidat kata penting yang sudah diurutkan

Kandidat Kata Penting	Skor
sumber cahaya halogen volt watt	23,6666
karakterisasi dye sensitized solar cell	18,5
dye sensitized solar cell	14,5
sel surya berbasis fotoelektrokimia	14
mengkonversi cahaya matahari	9,6666
penyerap cahaya matahari	9,6666
zat warna organik	9,0
buah naga merah	9,0
karakterisasi pengukuran tegangan	8,6666
separasi muatan listrik	8,0
sel surya	6,0
transfer muatan	4,5
energi listrik	4,5
dye sensitizer	4,5
variasi dye	4,5
semikonduktor anorganik	4,0
iodin solution	4,0
hylocereus polyrhizus	4.0
variasi elektrolit	3,4
Dye	2,5
Tegangan	1,6666
Elektrolit	1,4
Dssc	1.0
fabrikasi	1.0
aquades	1.0
ekstraksi	1.0
ketinggian	1.0
stabil	1.0
Arus	1.0

Tabel 4.9 Contoh pengambilan 4 kata penting

Kata Penting	Skor
sumber cahaya halogen volt watt	23,6666
karakterisasi dye sensitized solar cell	18,5
dye sensitized solar cell	14,5
sel surya berbasis fotoelektrokimia	14,0



Gambar 4.1 Ekstraksi kata penting menggunakan *Rapid Automatic Keyphrase Extraction*

4.3.2 Pembentukan Vektor Konsep

Setelah proses ekstraksi kata penting selesai, tahap selanjutnya yaitu proses pembentukan vektor konsep. Pada tahap ini akan dilakukan proses pembobotan menggunkan TF-IDF untuk medapatkan vektor konsep. Input dari proses pembobotan TF-IDF adalah kata penting dari proses *Rapid Automatic Keyphrase Extraction* (RAKE)

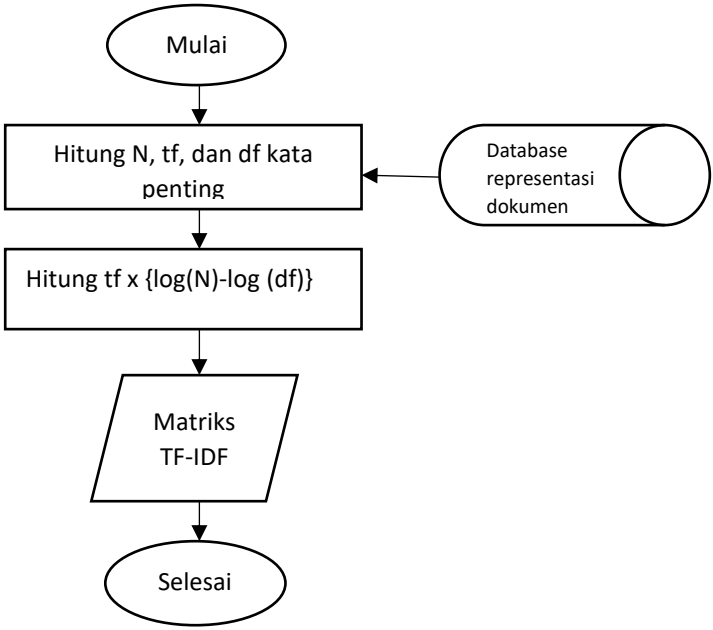
Langkah langkah pembentukan vektor konsep :

1. Hitung frekuensi kemunculan kata penting pada dokumen (*tf)*
2. Hitung jumlah koleksi dokumen yang ada (*N*)
3. Hitung jumlah dokumen yang mengandung kata penting tersebut (*df*)
4. Hitung $tf\ x\ \{\log(N) - \log(df)\}$ yang merupakan nilai TF-IDF setiap kata
5. Simpan seluruh konsep beserta matriks barisnya sebagai vektor konsep

Berikut merupakan contoh tampilan tabel pembobotan TF-IDF

Tabel 4.8 Contoh hasil TF-IDF

	1209100033	1209100076	1209100080	1210100027	...
agent model predictive control	0.0000	0.0000	0.0000	0.0000	...
agent mpc	0.0000	0.0000	0.0000	0.0000	...
alternating derection implicit	0.0000	0.0000	0.0000	0.0000	...
bahasa pemograman berorientasi objek	0.0000	1.0414	0.0000	0.0000	...
jenis finger print scanner	0.0000	1.0414	0.0000	0.0000	...
keamanan citra digital	1.0414	0.0000	0.0000	0.0000	...
...	...	...	...	...	



Gambar 4.1 Pembobotan TF-IDF

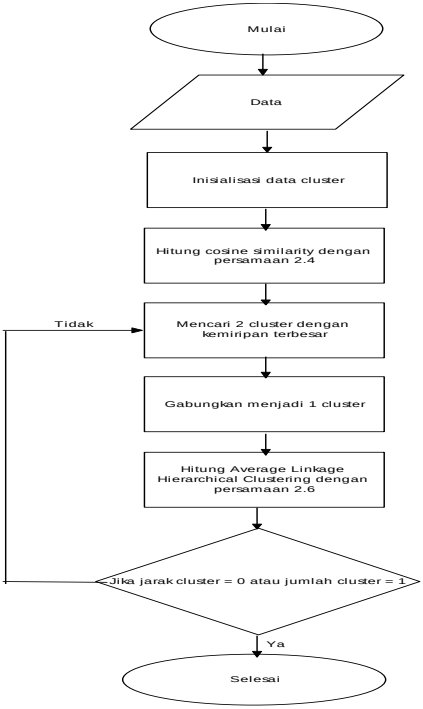
4.3.3 Pengklasteran

Proses pengklasteran dilakukan dengan menggunakan metode *clustering* yaitu *Average Linkage Hierarchical Clustering.* Proses ini bertujuan untuk mengelompokkan dokumen - dokumen ke dalam beberapa *cluster* dan menentukan topik pada *cluster*. Langkah – langkah pengklasteran sebagai berikut :

Input : Konsep beserta vektor baris hasil dari TF-IDF
Output : *Cluster* beserta topik pada *cluster*
Proses pengklasteran menggunakan *Average Linkage Hierarchical* :

1. Diasumsikan setiap dokumen sebagai *cluster*

2. Hitung jarak antar *cluster* dengan menggunakan *cosinus similarity* sesuai dengan Persamaan 2.4
3. Pilih 2 *cluster* yang memiliki jarak terbesar, kemudian gabungkan sehingga akan membentuk *cluster* baru.
4. Hitung jarak rata rata antar *cluster* dengan menggunakan *average linkage hierarchical clustering* sesuai persamaan 2.6
5. Ulangi langkah 3 dan 4 sampai jarak rata rata terbesar antar *cluster* sama dengan 0 atau jumlah *cluster* sama dengan 1



Gambar 4.2 Pengklasteran menggunakan *Average Linkage Hierarchical Clustering*

Setelah didapatkan kata penting pada setiap *cluster*, kemudian hitung frekuensi kata penting terbanyak untuk dijadikan topik *cluster.*

4.4 Implementasi Sistem

Setelah perancangan selesai, tahap selanjutnya adalah implementasi. Tahap ini bertujuan agar user dapat menggunakan program yang telah dirancang

4.4.1 Implementasi Interface

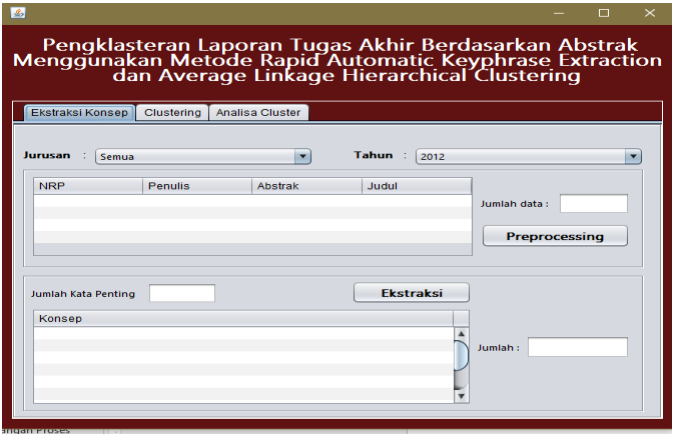
Pada sub bab ini dijelaskan tentang kegunaan fungsi – fungsi yang ada di dalam aplikasi beserta tampilan desaign. Antarmuka dibagi menjadi 3 tab yaitu Ekstraksi Konsep, *Clustering* dan *Analisa Cluster*.

1. Tab Ekstraksi Konsep

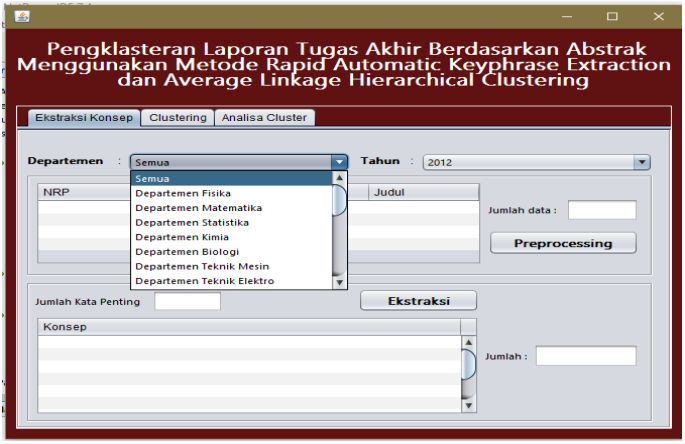
Pada tab ini terdapat menu departemen, menu tahun, tombol *preprocessing* dan tombol ekstraksi. Sebelum pengguna memulai proses *prepocessing*, pengguna harus memilih departemen yang diinginkan di menu departemen. Pada menu departemen pengguna dapat memilih semua departemen atau departemen yang diinginkan saja. Selain memilih departemen, pengguna juga harus memilih tahun untuk dapat menampilkan data sesuai dengan tahun yang telah dipilih. Pilihan tahun yang tersedia yaitu 2012, 2013, 2014, 2015 dan 2016. Pengguna dapat memilih data dalam *range* satu tahun atau secara keseluruhan yaitu 2012 sampai 2016. Setelah jurusan dan tahun dipilih, pengguna dapat melakukan proses *preprocessing* dengan menekan tombol *preprocessing*. Data yang telah dipilih oleh pengguna akan ditampilkan pada tabel, selain itu jumlah data juga akan di tampilkan pada kotak dialog jumlah data.

Setelah melalui tahap *prepocessing* data akan di ektraksi untuk mendapatkan konsep beserta vektor konsep. Pengguna harus memasukkan jumlah kata penting yang diinginkan pada field Jumlah Kata Penting sebelum melakukan proses

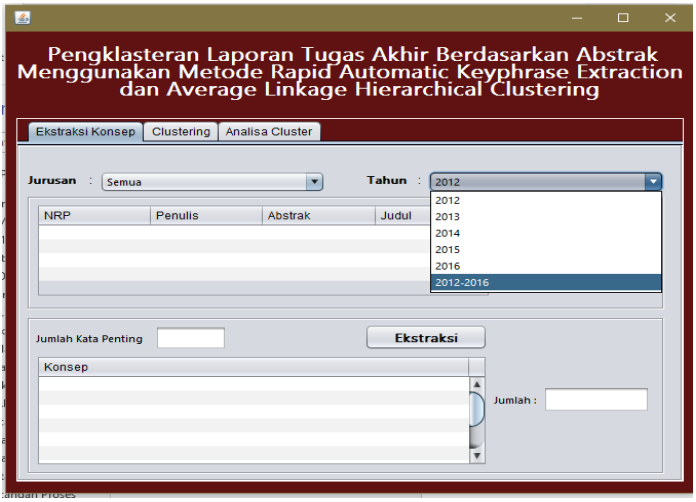
ekstraksi. Setelah proses ektraksi dilakukan, hasil ekstraksi akan ditampilkan pada tabel berupa konsep yang digunakan sebagai inputan pada tahap selanjutnya.



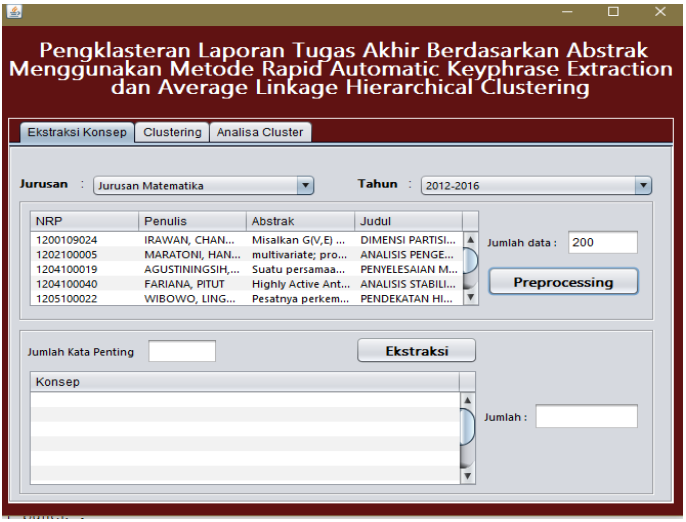
Gambar 4.3 Tab Ekstraksi Konsep



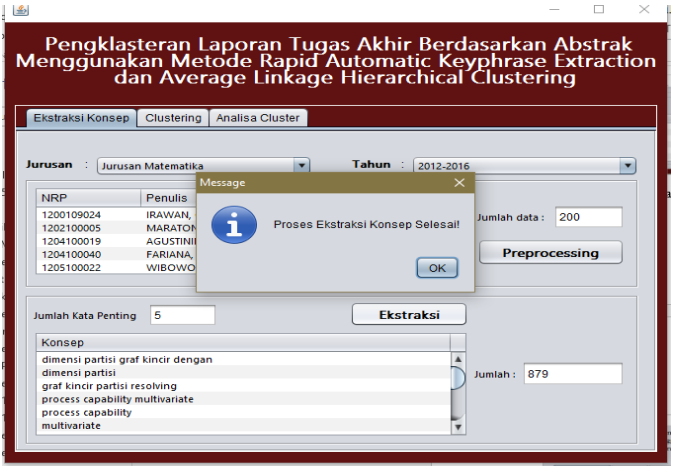
Gambar 4.4 Pilihan departemen pada combobox departemen



Gambar 4.5 Pilihan tahun pada combobox tahun



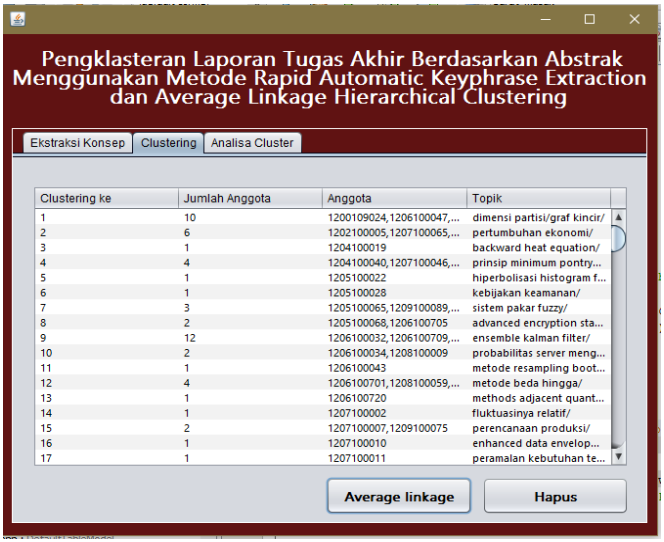
Gambar 4.6 *Preprocessing*



Gambar 4.7 Ekstraksi

2. Tab *Clustering*

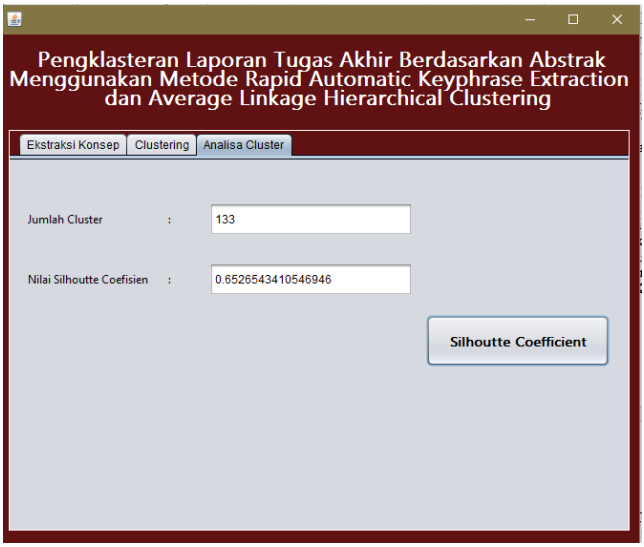
Tab ini merupakan tab yang digunakan untuk pengklasteran dan deteksi topik. Tab ini berfungsi untuk mengelompokkan konsep yang telah diekstraksi ke dalam beberapa *cluster*. Pada tahap ini pengguna harus menekan tombol Average Linkage untuk proses *clustering*. Hasil *cluster* yang ditampilkan pada proses ini yaitu *Clustering* ke-, jumlah angota, anggota dan topik yang dihasilkan.



Gambar 4.8 *Tab Clustering*

3. Tab Analisa *Cluster*

Tab ini merupakan tab yang berfungsi untuk menganalisa *cluster*. Pada tab ini pengguna dapat mengetahui apakah hasil *clustering* yang terbentuk sudah baik apa belum. Untuk mengetahui hasil *cluster* pengguna harus menekan tombol *Silhoutte Coefficient* terlebih dahulu.



Gambar 4.9 Tab Analisa *Cluster*

4.4.2 Implementasi Proses Ekstraksi Konsep

Proses ekstraksi konsep terdiri dari dua tahap yaitu ekstraksi kata penting dari abstraks dokumen dan pembentukan vektor konsep dengan menggunakan pembobotan TF-IDF. Ekstraksi kata penting dimulai dengan memisahkan abstraks dokumen berdasarkan tanda baca dan *stopword*. Proses ini dilakukan pada kelas RAKE. Kode yang digunakan untuk ekstraksi kata penting adalah

```
listKata = new ArrayList<>();
listKataPenting = new ArrayList<>();
databaseTA db = new databaseTA();
db.connectFirst();
String Dok = dokumen.getAbstrak();
ArrayList<String> tempKata = new ArrayList<>();
String hasil = prosesLowerCase(Dok);
/*Split berdasarkan tanda baca*/
ArrayList<String> kandidatKK = prosesSplitTB(hasil);
/*Split berdasarkan stopword*/
kandidatKK = prosesSplitStopword(kandidatKK);
```

Terdapat beberapa method yang digunakan pada kelas RAKE yaitu :

1. prosesLowerCase()
   Method ini berguna untuk mengubah seluruh huruf dalam abstrak menjadi huruf kecil. Berikut kode yang digunakan

```
 private String prosesLowerCase(String abstrakDok){

 abstrakDok = abstrakDok.toLowerCase();

 return abstrakDok;
       }
```

2. prosesSpiltTB()
   Method ini berguna untuk memisahkan abstrak berdasarkan tanda baca. Kode yang digunakan untuk memisahkan abstrak berdasarkan tanda baca adalah

```
    private ArrayList<String> prosesSplitTB(String
abstrakDok){

        ArrayList<String> hasil = new
ArrayList<String>();
        abstrakDok = abstrakDok.replaceAll("[0-9]", "");
        System.out.println("menghilangkan angka : " +
abstrakDok);
        String[] tempHasil =
abstrakDok.split("\\p{Punct}");
        for (String as : tempHasil){
            System.out.println("hasil punct :" +as);
        }
        for(int i = 0; i < tempHasil.length; i++){

        if(tempHasil[i].length() > 2){
            StringBuilder sb = new StringBuilder();
            String[] buildString = tempHasil[i].split("
");
            for(int j = 0; j < buildString.length; j++){
                if(buildString[j].length() > 2){
                    if(j < buildString.length-1){
                        sb.append(buildString[j]);
                        sb.append(" ");
```

```
                    }
                    else{
                        sb.append(buildString[j]);
                    }
                }
            }
            hasil.add(sb.toString());
        }
        }
        return hasil;
    }
```

3. prosesSplitStopword()

Method ini befungsi untuk memisahkan kandidat kata penting menurut *stopword.*

Langkah pertama pada tahap ekstraksi kata penting adalah membaca abstrak dokumen dari *database* dan mengubah seluruh *string* dalam abstrak tersebut menjadi huruf kecil. Langkah berikutnya adalah menghilangkan karakter angka dan memisahkan abstrak berdasarkan tanda baca dan menyimpannya ke dalam daftar *string.* Setelah proses pemisahan abstrak berdasarkan tanda baca, langkah berikutnya adalah memisahkan daftar *string* yang di hasilkan berdasarkan *stopword*. *String-string* tersebut selanjutnya disimpan sebagai kandidat kata penting. Berikut kode untuk proses pemisahan abstrak berdasarkan *stopword.*

```
private ArrayList<String> isStopword(String kata){
        ArrayList<String> splitKata2 = new
ArrayList<>();
        ArrayList<String> kataPenting = new
ArrayList<>();

            String[] splitKata = kata.split(" ");

            for(int i = 0; i < splitKata.length; i++){

                if(i == splitKata.length - 1){
                    if
(!listStopword.contains(splitKata[i])){
                    splitKata2.add(splitKata[i]);
```

```
                    StringBuilder sb = new
StringBuilder();
                    for(int j = 0; j <
splitKata2.size(); j++){
                        sb.append(splitKata2.get(j));
                      if(j != splitKata2.size() - 1 ){
                            sb.append(" ");
                       }
                    }
                    kataPenting.add(sb.toString());
                    splitKata2 = new ArrayList<>();
                    }else if
(listStopword.contains(splitKata[i])){
                    StringBuilder sb = new
StringBuilder();
                    for(int j = 0; j <
splitKata2.size(); j++){
sb.append(splitKata2.get(j));
if(j != splitKata2.size() - 1 ){
    sb.append(" ");
                  }
                }
kataPenting.add(sb.toString());
splitKata2 = new ArrayList<>();
                    }
                }
if(!listStopword.contains(splitKata[i])){
splitKata2.add(splitKata[i]);
System.out.println("1. " +splitKata[i]);
                }
else if (listStopword.contains(splitKata[i])) {
 StringBuilder sb = new StringBuilder();
        for(int j = 0; j < splitKata2.size(); j++){
             sb.append(splitKata2.get(j));
             if(j != splitKata2.size() - 1 ){
              sb.append(" ");
                       }
                    }
 kataPenting.add(sb.toString());
     System.out.println("2. "+ sb.toString());
       splitKata2 = new ArrayList<>();
                }
            }
           return kataPenting;
    }
```

Setelah pemisahan abstrak berdasarkan tanda baca dan *stopword* selesai dilakukan maka tahap berikutnya adalah menghitung nilai frekuensi, *degree*, dan rasio tiap kata. Tahap ini dimulai dengan memisahkan setiap kandidat kata penting berdasarkan karakter spasi (“ “) sehingga menjadi hanya satu kata. Seluruh kata tersebut disimpan ke dalam variabel daftar kata. Nilai frekuensi dihitung berdasarkan kemunculan sebuah kata pada daftar kata. Nilai *degree* dihitung berdasarkan kemunculan sebuah kata pada kandidat kata penting ditambah kemunculan kata tersebut pada daftar kata. Nilai rasio dihitung dengan cara membagi nilai *degree* dengan frekuensi kata tersebut. Kode untuk perhitungan nilai *degree*, frekuensi dan rasio sebagai berikut

```
/*Mencari frekuensi kata*/
        for(String s: tempKata){
            if(listKata.size() == 0){
                Kata katabaru = new Kata(s, 1, 0, 0);
                listKata.add(katabaru);
            }
            else{
                if(isAda(s)){
listKata.get(findIndex(s)).setFrek(listKata.get(findInd
ex(s)).getFrek()+1);
                }
                else{
        Kata katabaru = new Kata(s, 1, 0, 0);
                    listKata.add(katabaru);
                }
            }
        }
/*Mencari Degree*/
        for(String s: kandidatKK){
            int jumlah = 0;
            String[] split = s.split(" ");
            jumlah = split.length;
            if(jumlah>1){
                for(Kata k: listKata){
                    if(s.contains(k.getKata())){
                        k.setDeg(k.getDeg()+1);
                    }
                }
            }
        }
```

```
/*Menghitung rasio*/
        for(Kata k: listKata){
            k.setDeg(k.getDeg()+k.getFrek());
            k.setRasio(k.getDeg()/k.getFrek());
        }
```

Tahap berikutnya adalah mengukur nilai fitur kandidat kata penting. nilai fitur kandidat kata penting dihitung dengan menambahkan nilai rasio kata yang terdapat pada kandidat kata penting tersebut. Nilai-nilai tersebut kemudian diurutkan dari besar ke kecil. Setelah itu disimpan di dalam *database*. Berikut kode untuk proses perhitungan nilai fitur kandidat kata penting

```
/*Mencari Skor*/
        for(String s: kandidatKK){
            double skor = 0;
            for(Kata k:listKata){
                if(s.contains(k.getKata())){
                    skor += k.getRasio() ;
                }
            }
            KataPenting kk = new KataPenting(s, skor);
            listKataPenting.add(kk);
        }
```

Setelah seluruh kata penting dari seluruh dokumen berhasil diekstraksi maka tahap selanjutnya adalah ekstraksi konsep mengggunakan TF-IDF. Tahap ini dimulai dengan membentuk matriks kemunculan setiap kata penting pada dokumen. Berikut kode proses pembentukan matriks kemunculan kata penting

```
   public double[][] buildMatrix(){

            int numDoc = dokTAiDMap.size();
            int numKata penting = kata
pentingIDMap.size();
            double[][] data = new double[numKata
penting][numDoc];

            System.out.println(numDoc);
            System.out.println(numKata penting);
```

```
            System.out.println(dokKeyIDMap.size());

            //Membentuk matriks kemunculan tiap-tiap
kata penting pada tiap-tiap dokumen
            for(int i = 0; i < numKata penting; i++){
                for(int j = 0; j < numDoc; j++){
                    String dokName = dokTAiDMap.get(j);
                    Bag<String> dokKata penting =
dokKeyIDMap.get(dokName);
                    String kata penting = kata
pentingIDMap.get(i);
                    int df = dokKata
penting.getCount(kata penting);
                    data[i][j] = df;
                }
            }

            return data;
        }
```

Setelah frekuensi kemunculan kata penting pada tiap dokumen dihitung maka langkah selanjutnya adalah menghitung nilai *inverse document frequency (idf)* kata penting dan dilanjutkan menghitung nilai TF-IDF. Berikut kode menghitung TF-IDF

```
//Menghitung tf-idf tiap term pada matrix
        public Matrix tfidfIndexer(Matrix matrix){
            int n = matrix.getColumnDimension();
            for(int j = 0; j <
matrix.getColumnDimension(); j++){
                for(int i = 0; i <
matrix.getRowDimension(); i++){
                    double matrixElement =
matrix.get(i, j);
                    if(matrixElement > 0.0D){
                        double dm = countDocsWithWord(
                        matrix.getMatrix(i, i, 0,
matrix.getColumnDimension() - 1));//df
                        matrix.set(i, j, matrix.get(i,
j) * (Math.log10(n) - Math.log10(dm)));
                    }
                }
            }
```

4.4.3 Implementasi Proses Pengklasteran

Proses pengklasteran dilakukan pada *class averagelinkage*. Tahap awal yang dilakukan pada proses *clustering* yaitu menghitung nilai jarak (*similarity*) antar *cluster* menggunakan *cosine similarity*. Kode menghitung jarak dengan *cosinus similarity* adalah

```
public double getCosinus(double A[], double B[]){
        int length = A.length;
        double atas = 0;
        double bawah1 = 0;
        double bawah2 = 0;
        for(int i = 0; i < length; i++){
            atas = A[i] * B[i] + atas;
            bawah1 = A[i] * A[i] + bawah1;
            bawah2 = B[i] * B[i] + bawah2;
        }
        double bawah = Math.sqrt(bawah1) *
Math.sqrt(bawah2);
        double similarity = atas/bawah;
        return (similarity);
    }
```

Setelah perhitungan jarak antar *cluster* didapat, tahap selanjutnya, mencari 2 *cluster* terdekat (paling mirip) yaitu dengan mencari *similarity* terbesar antar *cluster* dan menggabungkannya. Tahap selanjutnya yaitu menghitung jarak (*similarity*) rata – rata antar *cluster* menggunakan *average linkage hierarchical clustering*. Berikut kode untuk pengklasteran dengan *average linkage hierarchical clustering.*

```
public static double[][] getMatrixBaru(double[][] data,
int data1, int data2, int jumlah1, int jumlah2){
        int row = data.length;
        double[][] hasil = new double[row-1][row-1];
        int m = 0;
        for(int i = 0; i < row; i++){
            int k = 0;
            for(int j = 0; j < row; j++){
                if(j==data2){
                    k = j-1;
                }
                else if(i==data2){
```

```
                    m = i-1;
                }
                else{
                    hasil[m][k] = data[i][j];
                }
                k++;
            }
            m++;
        }

        int l = 0;
        for(int i = 0; i < row; i++){
            if(i!=data2){
                //buat masukkin rata - rata jarak
                hasil[data1][l]                                    =
(jumlah1*data[data1][i]                                            +
jumlah2*data[data2][i])/(jumlah1+jumlah2);
                hasil[l][data1] = hasil[data1][l];
            }
            else{
                l = i-1;
            }
            l++;
        }

        for(int i = 0; i < row-1; i++){
            hasil[i][i] = 0;
        }
        return hasil;
    }
```

Setelah *cluster* terbentuk, tahap selanjutnya menentukan topik pada setiap *cluster*. Topik di ambil dari frekuensi kemunculan kata penting terbanyak. Berikut kode penetuan topik.

```
ArrayList <String> list = new ArrayList <String> ();
            for(int i = 0; i < NRPasli.length; i++){
               String[] bagian= NRPasli[i].split("/");
                list.add(bagian[0]);
                  }
        String[] array = new String[list.size()];
        for (int i=0; i<list.size(); i++){
            array[i]=list.get(i);
        }
         TOPIK = new String[jumlahCluster];
```

```
        int lokasi = 0;
        boolean ada = true;
        int k = 0;
                double besar = 0.0;
                int posisi = 0; //memunculkan cuman satu
topik
        for(int i = 0; i < jumlahCluster; i++){
            int[] irisan = new int[kataKunci.length];
            for(int j = 0; j < irisan.length; j++){
                irisan[j] = 0;
            }
            System.out.println("Cluster "+(i+1));
            String[]              parts              =
NRPkatakunci[i].split("/");
            System.out.print("NRP \t\t :");
            for(int j = 0; j < parts.length; j++){
                System.out.print(parts[j]+", ");
                lokasi = 0;
                ada = true;
                k = 0;
                besar = 0.0;
                while(ada && k < array.length){
                    if(parts[j].compareTo(array[k])  ==
0){
                        ada = false;
                        lokasi = k;
                    }else
                        k++;
                }

                for(int l = 0; l < kataKunci.length;
l++){
                   if(data[l][lokasi]!=0.0){
                        irisan[l]++;
                        if(data[l][lokasi] > besar){
                            besar = data[l][lokasi];
                            posisi = l; //memunculkan
cuman satu topik
                        }
                    }
                }
            }
            int max = 0;
            for(int j=0; j<irisan.length; j++){
                if(irisan[j] > max){
                    max = irisan[j];
                }
            }
            System.out.println("");
```

```
            System.out.print("Topik \t\t :");
            String simpantopik="";
            for(int j = 0; j < kataKunci.length; j++){
                if(parts.length == 1){
System.out.print(kataKunci[j]+"/ ");
simpantopik= simpantopik+kataKunci[j]+"/";
                    }
                }
                else if(irisan[j] == max){
                    System.out.print(kataKunci[j]+"/
");
                     simpantopik=
simpantopik+kataKunci[j]+"/";
                }
            }
           System.out.println("");
           TOPIK[i]=simpantopik;

           setNrpCluster(NRPkatakunci);
       }
    }
```

Proses selanjutnya yaitu evaluasi *clustering* dengan menggunakan *silhoutte coefficient*. Kode untuk evaluasi *clustering* dengan *silhoutte coefficient* adalah

```
     silhoutte(Matrix matrix, String[] nrp, String[]
nrpCluster) {
        this.nrpCluster = nrpCluster;
        this.nrp = nrp;

        for(int i = 0; i < nrp.length; i++){
            nrp[i] = nrp[i].substring(0, 10);
        }

        posisiCluster = new int[nrp.length];
        jumlahAnggota = new int[nrpCluster.length];
        for(int i = 0; i < nrpCluster.length; i++){
            jumlahAnggota[i] = 0;
        }

        for(int i = 0; i < nrp.length; i++){
            for(int j = 0; j < nrpCluster.length; j++){
                if(nrpCluster[j].contains(nrp[i])){
                    posisiCluster[i] = j;
                    jumlahAnggota[j]++;
                }
```

```
            }
        }

        hitungSiluet                =                new
double[nrp.length][nrpCluster.length];
        for(int i = 0; i < nrp.length; i++){
            for(int j = 0; j < nrpCluster.length; j++){
                hitungSiluet[i][j] = 0;
            }
        }

        nilaiSiluet = 0;

        //this.lstKata pentings=lstKata pentings;
        int m = matrix.getRowDimension();
        int n = matrix.getColumnDimension();
        this.data = new double[m][n];
        for(int i = 0; i < m; i++){
            for(int j = 0; j < n; j++){
                data[i][j] = matrix.get(i, j);
            }
        }

        int col = data[0].length;
        int row = data.length;

        jarakeuclidian = new double[col][col];

        double min = 999;
        int data1 = 0; int data2 = 0;

        double[][] transpose = new double[col][row];
        for(int i = 0; i < row; i++){
            for(int j = 0; j < col; j++){
                transpose[j][i] = data[i][j];
            }
        }

        for(int i = 0; i < col; i++){
            for(int j = 0; j < col; j++){
                jarakeuclidian[i][j]    =    getDistance
(transpose[i], transpose[j]);
            }
       }

        //hitung matrix untuk siluet
        for(int i = 0; i < nrp.length; i++){
            for(int j = 0; j < nrp.length; j++){
```

```
                hitungSiluet[i][posisiCluster[j]]     =
hitungSiluet[i][posisiCluster[j]]                     +
(jarakeuclidian[i][j]/jumlahAnggota[posisiCluster[j]]);
            }
        }

        System.out.println("Matriks hitung Siluet");
        for(int i = 0; i < nrp.length; i++){
            for(int j = 0; j < nrpCluster.length; j++){
                System.out.print(hitungSiluet[i][j]   +
"/t");
            }
            System.out.println("");
        }

        double a = 0;
        double b = 999;

        for(int i = 0; i < nrp.length; i++){
            for(int j = 0; j < nrpCluster.length; j++){
                if(posisiCluster[i] == j){
                    a = hitungSiluet[i][j];
                } else{
                    if(hitungSiluet[i][j] < b){
                        b = hitungSiluet[i][j];
                    }
                }
            }
            System.out.println("a = "+a);
            System.out.println("b = "+b);
            nilaiSiluet   =   nilaiSiluet   +   ((b   -
a)/Math.max(a, b));
            b = 999;
            totalSiluet = nilaiSiluet/nrp.length;
            }
         System.out.println("Silhouette  Coefficient  =
"+totalSiluet);
         System.out.println("nrp length"+nrp.length);
    }
```

BAB V
HASIL DAN PEMBAHASAN

Pada bab ini dijelaskan tentang hasil uji coba dan pembahasan dari program yang telah dibuat.

5.1 Data Uji Coba

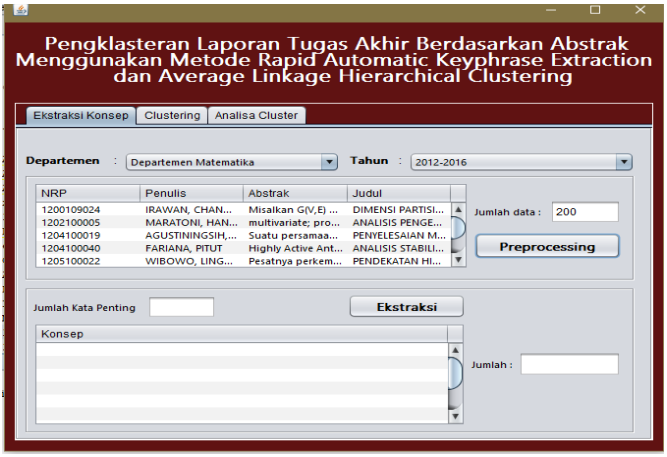
Data yang digunakan dalam Tugas Akhir ini adalah data Tugas Akhir mahasiswa ITS pada tahun 2012 hingga 2016. Terdapat 6.916 data Tugas Akhir. Data disimpan dalam database MySQL. Dari 6.916 data, dilakukan percobaan terhadap 607 data dari departemen Fisika, departemen Matematika dan departemen Teknik Perkapalan. Berdasarkan perangkat lunak dan perangkat keras yang digunakan, data yang mampu diproses sebanyak 1.800 data dari 6.916 data. Pada pengelohan data diambil beberapa field pada tabel yaitu NRP, abstrak, departemen dan tahun.

5.2 Pengelompokan Konsep

Uji pengelompokan konsep bertujuan untuk mengelompokkan konsep yang memiliki kemiripan topik ke dalam beberapa *cluster*. Kemampuan mengelompokan dilakukan dengan menghitung jarak rata rata nilai kemiripan topik terhadap seluruh koleksi dokumen.

Tahap yang perlu dilakukan sebelum pengelompokan konsep adalah proses *prepocessing* data dan ekstraksi konsep. Pada tahap *preprocessing*, dilakukan proses *case folding* dan *filtering*. *Case folding* adalah merubah semua huruf dalam dokumen menjadi huruf kecil dan menghilangkan karakter selain huruf sedangkan *filtering* adalah menghilangkan kata kata yang tidak penting yang ada pada *database stopword*. Sebelum menekan tombol *preprocessing* pengguna harus memilih departemen serta tahun yang diinginkan. Uji coba dilakukan pada departemen Fisika, departemen Matematika dan departemen Teknik Perkapalan pada tahun 2012-2016.

Data yang diolah pada departemen Fisika sebanyak 192 data, 200 data pada departemen Matematika dan 215 data pada departemen Teknik Perkapalan.



Gambar 5.1 *Preprocessing* Data pada departemen matematika tahun 2012-2016

Setelah tahap *preprocessing* selesai, tahap selanjutnya yaitu proses ekstraksi kata penting menggunakan *Rapid Automatic Keyphrase Extraction* dan ekstraksi konsep menggunakan TF-IDF. Pada tahap ini, dilakukan uji coba untuk departemen Fisika, departemen Matematika dan departemen Teknik Perkapalan sebanyak 3 kali dengan memilih jumlah kata penting secara acak yaitu 2, 4 dan 6.

Hasil ekstraksi konsep dengan memasukkan 2 kata penting menghasilkan 373 kata penting pada departemen Fisika, 394 kata penting pada departemen Matematika dan 422 kata penting pada departemen Teknik Perkapalan. Sementara hasil ektraksi konsep dengan memasukkan 4 kata penting yaitu 728 kata penting pada departemen Fisika, 772 kata penting

pada departemen Matematika dan 833 kata penting pada departemen Teknik Perkapalan.

Hasil ekstraksi konsep dengan memasukkan 6 kata penting menghasilkan kata penting yang lebih besar dibandingkan dengan hasil ektraksi konsep dengan menggunakan 2 dan 4 kata penting. Dengan memasukkan 6 kata penting, hasil ektraksi konsep yang dihasilkan yaitu 1.077 pada departemen Fisika, 1.137 pada departemen Matematika dan 1.243 kata penting pada departemen Teknik Perkapalan. Berikut tabel hasil uji coba dengan jumlah kata penting yang dimasukkan adalah 2, 5, dan 7 :

Tabel 5. 1 Uji coba dengan jumlah kata penting = 2

Departemen	Jumlah Data	Total Kata Penting yang Dihasilkan
Fisika	192	373
Matematika	200	394
Teknik Perkapalan	215	422

Tabel 5. 2 Uji coba dengan jumlah kata penting = 4

Departemen	Jumlah Data	Total Kata Penting yang Dihasilkan
Fisika	192	728
Matematika	200	772
Teknik Perkapalan	215	833

Tabel 5. 3 Uji coba dengan jumlah kata penting = 6

Departemen	Jumlah Data	Total Kata Penting yang Dihasilkan
Fisika	192	1.077
Matematika	200	1.137
Teknik Perkapalan	215	1.243

Berdasarkan Tabel 5.1, Tabel 5.2 dan Tabel 5.3 jumlah kata penting yang dimasukkan/diinputkan akan mempengaruhi total kata penting yang dihasilkan. Semakin besar kata penting yang dimasukkan maka semakin besar juga total kata penting yang dihasilkan. Namun apabila kata penting yang diinginkan kecil maka jumlah kata penting yang dihasilkan juga kecil.

5.3 Pengklasteran dan Penentuan Topik

Hasil dari ektraksi konsep akan digunakan pada tahap selanjutnya yaitu *clustering*. *Clustering* dilakukan untuk mengelompokan dokumen yang memiliki kemiripan. Pada penelitian ini penulis menggunakan ukuran jarak *cosinus similarity*. Nilai yang dihasilkan *cosine similarity* adalah 0 sampai dengan 1. Jika nilai *cosine similarity* semakin mendekati 1, menunjukan dokumen mempunyai tingkat kemiripan yang tinggi, namun apabila nilai *cosinus similarity* mendekati 0 maka dokumen tersebut mempunyai tingkat kemiripan yang rendah atau tidak mirip sama sekali. Setelah perhitungan *cosinus similarity* selesai dilakukan *clustering* menggunakan *Average Linkage Hierarchical Clustering*. Pada penelitian ini, *cluster* akan berhenti ketika jarak(*similarity*) rata rata terbesar bernilai 0 atau ketika jumlah *cluster* sama dengan satu. Hal ini dikarenakan, apabila jarak(*similarity*) rata rata sama dengan 0 menandakan dokumen tidak memiliki kemiripan sama sekali. Setelah *cluster* didapatkan, maka akan ditampilkan topik yang merepresentasikan *cluster* tersebut. Topik didapatkan dari kata penting yang memiliki frekuensi kemunculan terbanyak pada *cluster*. Berikut akan dijelaskan hasil *clustering* berdasarkan jumlah kata penting.

5.3.1 Hasil *clustering* dengan 2 Kata Penting

Dari uji coba *clustering* dengan pengambilan 2 kata penting menghasilkan 155 *cluster* pada departemen Fisika, 169 *cluster* departemen Matematika dan 179 *cluster* pada departemen Teknik Perkapalan. Jumlah anggota terbanyak

yang dihasilkan untuk departemen Fisika adalah 6 yang terletak di *cluster* ke 18 dengan topik yaitu *dye sensitized solar cell*, sementara jumlah anggota terbanyak di departemen Matematika adalah 6 yang terletak di *cluster* ke 1 dengan topik yaitu dimensi partisi/graf kincir, dan jumlah anggota terbanyak untuk departemen Teknik Perkapalan adalah 14 anggota yang terletak di *cluster* ke 4 dengan topik yaitu proses pengelasan. Berikut tabel hasil uji coba *clustering* dengan 2 kata penting :

Tabel 5. 4 Hasil *clustering* dengan jumlah kata penting = 2

Departemen	Jumlah Cluster	Anggota terbanyak	Cluster ke-	Topik
Fisika	155	6	18	dye sensitized solar cell
Matematika	169	6	1	dimensi partisi/ graf kincir
Teknik Perkapalan	179	14	4	proses pengelasan

5.3.2 Hasil *clustering* dengan 4 kata penting

Uji coba *clustering* dengan pengambilan 4 kata penting menghasilkan 99 *cluster* pada departemen Fisika, 116 *cluster* departemen Matematika dan 127 *cluster* pada departemen Teknik Perkapalan. Jumlah anggota terbanyak yang dihasilkan untuk departemen Fisika adalah 39 yang terletak di cluster ke 2 dengan topik yaitu *scanning electron microscopy*, sementara jumlah anggota terbanyak di departemen Matematika adalah 33 yang terletak di *cluster* ke 3 dengan topik yaitu metode beda hingga, dan jumlah anggota terbanyak untuk departemen Teknik Perkapalan adalah 72 anggota yang terletak di *cluster* ke 1 dengan topik yaitu metode elemen hingga. Berikut tabel hasil uji coba *clustering* dengan 4 kata penting :

Tabel 5. 5 Hasil *clustering* dengan jumlah kata penting = 4

Departemen	Jumlah cluster	Anggota terbanyak	Cluster ke -	Topik
Fisika	99	39	2	scanning electron microscopy
Matematika	116	33	3	metode beda hingga
Teknik Perkapalan	127	72	1	metode elemen hingga

5.3.3 Hasil *clustering* dengan 6 kata penting

Uji coba *clustering* dengan pengambilan 6 kata penting menghasilkan 77 *cluster* pada departemen Fisika, 87 *cluster* departemen Matematika dan 99 *cluster* pada departemen Teknik Perkapalan. Jumlah anggota terbanyak yang dihasilkan untuk departemen Fisika adalah 46 yang terletak di *cluster* ke 2 dengan topik yaitu scanning electron microscopy, sementara jumlah anggota terbanyak di departemen Matematika adalah 52 yang terletak di *cluster* ke 1 dengan topik metode beda hingga, dan jumlah anggota terbanyak untuk departemen Teknik Perkapalan adalah 97 anggota yang terletak di *cluster* ke 1 dengan topik yaitu metode elemen hingga. Berikut tabel hasil uji coba *clustering* dengan 6 kata penting :

Tabel 5. 6 Hasil *clustering* dengan jumlah kata penting = 6

Jurusan	Jumlah *cluster*	Anggota terbanyak	*Cluster* ke-	Topik
Fisika	77	46	2	scanning electron microscopy
Matematika	87	52	1	metode beda hingga
Teknik Perkapalan	99	97	1	metode elemen hingga

Dari Tabel 5.4, Tabel 5.5 dan Tabel 5.6 dapat dilihat bahwa banyaknya kata penting, akan mempengaruhi jumlah *cluster* yang dihasilkan. Ketika jumlah kata penting yang dimasukkan semakin banyak maka jumlah *cluster* yang dihasilkan akan semakin sedikit.

5.4 Analisa Hasil *Cluster*

Dokumen-dokumen yang sudah terkelompokkan dalam satu *cluster* akan dianalisa kesamaan topiknya antara dokumen satu dengan dokumen yang lain. Analisa dilakukan pada hasil *clustering* departemen Fisika dan departemen Matematika dengan jumlah kata penting adalah 2.

a. Departemen Fisika

   Dari hasil uji coba yang telah dilakukan sebelumnya, dokumen abstrak yang bergabung pada *cluster* ke 18 mempunyai anggota sebanyak 6. Berikut daftar dokumen abstrak tersebut :

Tabel 5. 7 Dokumen abstrak departemen Fisika *cluster* ke 18

NRP	Abstrak
1108100008	Telah dilakukan studi awal fabrikasi dan karakterisasi **dye sensitized solar cell** (DSSC) menggunakan kulit manggis (Garcinia mangostana) sebagai dye sensitizer dengan metode spin coating dalam pelapisan TiO2. Variasi kecepatan dan lama pemutaran daripada spin coating dilakukan untuk mengetahui pengaruh terhadap nilai arus dan tegangan yang di hasilkan oleh **dye sensitized solar cell** (DSSC) . Metode penelitian dilakukan dengan cara pembuatan prototype **dye sensitized solar cell** (DSSC) yang kemudian di sinari dengan lampu halogen sebagai sumber cahaya. Berdasarkan

	penelitian yang telah dilakukan di dapatkan bahwa semakin besar kecepatan putarnya akan semakin besar nilai arusnya. Sedangkan, Untuk lama pemutaran hanya berpengaruh terhadap kehomogenan lapisan TiO2.
1108100017	Telah dilakukan studi pendahuluan fabrikasi dan karakterisasi **Dye Sensitized Solar Cell** (DSSC) menggunakan ekstraksi daun bayam (Amaranthus Hybridus L.) sebagai dye sensitizer. **Dye Sensitized Solar Cell** (DSSC) merupakan sel surya yang dapat mengkonversi energi foton menjadi energi listrik. **Dye Sensitized Solar Cell** (DSSC) dibentuk dengan struktur sandwich dimana terdapat empat bagian antara lain : Kaca ITO (Indium Tin Oxide) sebagai substrat; TiO2 sebagai bahan semikonduktor; Dye alami sebagai donor elektron; Elektrolit sebagai transfer elektron. Penelitian dilakukan dengan mengukur arus dan tegangan terhadap waktu dengan variasi sumber cahaya matahari dan lampu halogen dengan perbedaan jarak ketinggian terhadap sel **Dye Sensitized Solar Cell** (DSSC). Pengujian menggunakan sumber cahaya matahari lebih besar daripada menggunakan lampu halogen. Tegangan dan arus dari lampu halogen dengan ketinggian 5cm terhadap **Dye Sensitized Solar Cell** (DSSC) lebih besar daripada pada ketinggian 20cm dan 35cm. Hasil ini memperlihatkan bahwa jarak menentukan intensitas lampu halogen yang diterima oleh sel. Semakin tinggi jarak lampu halogen terhadap sel, semakin kecil

	intensitas, dan semakin kecil nilai arus dan tegangan.
1108100023	Telah dilakukan fabrikasi dan karakterisasi **Dye Sensitized Solar Cell** (DSSC) dengan menggunakan ekstraksi daging buah naga merah sebagai dye sensitizer. **Dye Sensitized Solar Cell** (DSSC) merupakan sel surya berbasis fotoelektrokimia dimana digunakan zat warna organik sebagai penyerap cahaya matahari dan semikonduktor anorganik sebagai tempat terjadinya separasi muatan listrik. **Dye Sensitized Solar Cell** (DSSC) dapat mengkonversi cahaya matahari menjadi energi listrik dengan menggunakan elektrolit sebagai transfer muatan. Penelitian dilakukan dengan variasi dye dan variasi elektrolit pada ketinggian 5 cm dan 10 cm. Dilakukan karakterisasi pengukuran tegangan dan arus terhadap waktu dengan sumber cahaya halogen 6 volt 30 watt. Dari hasil pengujian diperoleh tegangan dan arus yang lebih tinggi dan stabil pada **Dye Sensitized Solar Cell** (DSSC) dengan dye(100 gr daging buah naga merah+5 ml aquades) dari pada DSSC dengan dye(100 gr daging buah naga merah+10 ml aquades). Sedangkan pada variasi elektrolit, tegangan dan arus yang dihasilkan oleh **Dye Sensitized Solar Cell** (DSSC) dengan elektrolit(6 gr KI+3 ml iodin solution 10%) lebih tinggi dan stabil dari pada **Dye Sensitized Solar Cell** (DSSC) dengan elektrolit (3 gr KI+3 ml iodin solution 10% dan 3 gr KI+6 ml iodin solution 10%).

1109100005	**Dye Sensitized Solar Cell** (DSSC) dengan substrat kaca Fluorine doped Tin Oxide (FTO) dan lapisan oksida berupa TiO2 nanosize yang disensitasi â€□dyeâ€□ ekstrak ekulit luar buah Manggis telah dibuat dengan metode spin coating. Elektrolit gel digunakan sebagai media transfer elektron dengan menambahkan polimer Polyethilene Glicol (PEG) 1000. TiO2 nanosize dibuat dengan metode kopresipitasi dari larutan TiCl3. Identifikasi fasa dan langkah awal identifikasi ukuran kristal TiO2 menggunakan analisa data hasil uji difraksi kristal TiO2 dengan Diffractometer Sinar-X. Tahap lanjut identifikasi ukuran kristal menggunakan software Materials Analysis Using Diffraction (MAUD) menghasilkan ukuran kristal TiO2 sebesar 10.5 nm Penentuan jenis fasa TiO2 dilakukan dengan software Match dan menunjukan bahwa fasa TiO2 yang terbentuk adalah anatase. Sedangkan uji absorbansi ekstrak kulit luar buah manggis menggunakan Spectrofotometri UV-Vis menunjukkann bahwa ekstrak kulit luar buah manggis mampu mengabsorb foton pada daerah Near Infra Red (NIR) dan daerah blue-yellow dari visible light. Penggunaan TiO nano sebagai lapisan oksida terbukti mampu mencapai arus short circuit sebesar 30.9 ÂµA, tegangan open circuit sebesar 398.3 mV dan kestabilan penggunaan yang lama .
1107100011	Telah dilakukan penelitian Tugas Akhir yang berjudul â€œPembuatan dan Karakterisasi Prototipe **Dye Sensitized Solar Cell** (DSSC)

	Menggunakan Ekstraksi Kulit Buah Manggis Sebagai Dye Sensitizerâ€• dengan variasi komposisi penyusun elektrolit 3 gram KI dan 3 ml Iodine, 3 gram KI dan 6 ml Iodine, 6 gram KI dan 3 ml Iodine, dan 9 gram KI dan 3 ml Iodine. Selain itu diberikan juga variasi pada suhu sintering pada lapisan TiO2 sebesar 300Â° C dan 400Â° C. **Dye Sensitized Solar Cell** (DSSC) ini dianalisa dengan menggunakan sumbar cahaya lampu halogen. Penelitan ini juga dilakukan karakterisasi pada dye kulit buah manggis dengan menggunakan alat spektrofotometer UV-Vis. Hasil penelitian Tugas Akhir ini adalah dapat dibuatnya prototipe DSSC yang dapat menghasilkan arus dan tegangan, hasil karakterisasi dye kulit buah manggis dengan menggunakan alat spektrofotometer UV-Vis, dan didapatkan nilai tegangan dan arus lebih besar pada DSSC yang diberi suhu sintering pada lapisan TiO2 400Â° C dibanding yang menggunakan suhu 300Â° C.
1108100027	Telah dilakukan penelitian mengenai pengaruh pemberian space (bantalan) untuk mendapatkan kestabilan arus dan tegangan prototipe DSSC (**Dye Sensitized Solar Cell**) dengan ekstraksi kulit buah manggis (Garcinia mangostana L.) sebagai dye sensitizer. Penelitian ini bertujuan untuk membuat prototipe DSSC (**Dye Sensitized Solar Cell**) yang dapat mengkonversi energi cahaya menjadi energi listrik dengan TiO2 sebagai bahan semikonduktor, serta mengetahui pengaruh dari pemberian space (bantalan) pada penyusunan lapisan DSSC

	(**Dye Sensitized Solar Cell**) terhadap arus, tegangan dan rentang waktu kestabilan yang dihasilkan. Pengujian dilakukan menggunakan lampu halogen dengan jarak sel 5cm dari lampu. DSSC (**Dye Sensitized Solar Cell**) dengan bantalan terbukti lebih stabil dalam arus dan tegangan serta memiliki rentang ketahanan yang lebih lama. Efisiensi DSSC (**Dye Sensitized Solar Cell**) dengan space (bantalan) didapatkan nilai sebesar 1,42%.

Proses pengklasteran diawali dengan bergabungnya dokumen yang memiliki NRP 1108100008 dengan dokumen yang memiliki NRP 1108100017 dengan jarak(*similarity)*nya adalah 0,89406. Setelah kedua dokumen tersebut bergabung menjadi satu *cluster*, dokumen dengan NRP 1108100023 bergabung dengan *cluster* tersebut dengan jarak (*similarity*) nya yaitu 0,724871. Selanjutnya penggabungan terjadi pada dokumen yang mililiki NRP 1107100011 dengan dokumen yang memiliki NRP 1108100027 dengan jarak (*similarity*) 0,408675. Proses selanjutnya yaitu dokumen dengan NRP 1109100005 bergabung dengan *cluster* yang beranggotakan NRP 1108100008, 1108100017, 1108100023 dengan jarak(*similarity*) 0,34145. Kemudian semua dokumen bergabung dengan jarak (*similarity*) 0,18472 sehingga terbentuk dalam satu *cluster*.

Setelah terbentuk menjadi satu *cluster*, ditentukan apa topik *cluster* tersebut. Topik didapatkan dari kata penting yang memiliki frekuensi kemunculan terbanyak pada *cluster*. Berdasarkan Tabel 5.7 semua dokumen membahas mengenai *Dye Sensitized Solar Cell***.** Sehingga pokok bahasan / topik yang memenuhi adalah **Dye Sensitized Solar Cell.**

b. Departemen Matematika

Dari hasil uji coba yang telah dilakukan sebelumnya, dokumen abstrak yang bergabung pada *cluster* ke-1 mempunyai anggota sebanyak 6 anggota. Berikut daftar dokumen abstrak tersebut :

Tabel 5.8 Dokumen abstrak departemen Matematika *cluster* ke 1

NRP	Abstrak
1200109024	Misalkan G(V,E) adalah graf terhubung dan S adalah sebuah subset dari V(G), jarak antara v dan S adalah. Suatu graf terhubung G dengan k buah partisi dari V(G) dan v vertex di G, representasi v pada adalah resolving dari V(G), jika k adalah nilai minimum sedemikian hingga adalah partisi resolving dari graf G, maka k adalah **dimensi partisi** dari graf G,atau ditulis pd = k. Dalam tugas akhir ini akan dibuktikan **dimensi partisi graf Kincir** dengan m-bilah = k sedemikian hingga. Partisi resolving; **Dimensi partisi**; **Graf kincir**.
1206100047	Graf adalah himpunan pasangan terurut dari himpunan simpul dan sisi yang dinotasikan dengan G=(V,E) dimana V adalah himpunan simpul dan E adalah himpunan sisi yaitu himpunan pasangan simpul dari V. Jika graf terhubung G, Untuk S ⊆ V(G) dan simpul v ∈V(G), jarak antara v dengan S yang dinotasikan sebagai d(v,S) dan didefinisikan sebagaid(v,S)=min⁡〖{d(v,x) ┤|x∈ S}.〗

	Untuk h-partisi terurut Π = {S1, S2,..., Sh} pada V(G) dan simpul v∈V(G), representasi dari v terhadap Π didefinisikan sebagai h-vektor ditulis r(v|Π) = (d(v,S1), d(v,S2), d(v,S3),..., d(v,Sh)). Jika h-vektor r(v| Π), untuk setiap simpul v pada V(G) berbeda, maka Π disebut partisi pembeda dari V(G). Nilai h-minimum dimana terdapat h-partisi pembeda dari V(G) disebut **dimensi partisi** dari G dinotasikan dengan pd(G). Pada Tugas Akhir ini ditentukan **dimensi partisi** pada pengembangan **graf kincir** dengan pola K1 + mCn dengan m≥2, n≥3. Dari analisis yang dilakukan, diperoleh hasil bahwa **dimensi partisi** K1 + mCn , n≥3,m≥2, dengan m dan n bilangan bulat positif adalah: pd(K_1+mC_n)=(4, untuk n=4 dan 2≤m≤3@ n=5 dan m=2@@5,untuk n=4 dan 4≤m≤5@ n=5 dan m=3)┤. Dan batas atas pd(K_1+mC_n)≤h dengan h bilangan bulat terkecil yang memenuhi (■(h@n))≥m, untuk h≥6 dan n, m yang lain. partisi pembeda; **dimensi partisi**; **graf kincir**.
1206100015	Graf adalah himpunan pasangan (V,E) dimana V adalah himpunan hingga tidak kosong simpul (vertex) dan E adalah himpunan sisi (edge) yaitu pasangan simpul dari V. Jika G adalah graf

	terhubung, misalkan S ⊆ V(G) dan simpul v ∈ V(G), jarak antara v dengan S adalah d(v,S) dengan d(v,S) = min{d(v,x)| x ∈ S}. Misalkan k buah partisi dan untuk himpunan terurut Π = {S1, S2, ..., Sk} dari simpul-simpul dalam graf terhubung G dan simpul v pada V(G), representasi dari v terhadap Π adalah r(v|Π) dengan r(v|Π) = (d(v,S1), d(v,S2), ..., d(v,Sk)). Jika k-vektor r(v|Π), untuk setiap simpul v pada V(G) berbeda, maka Π disebut himpunan partisi pembeda dari V(G). Bilangan bulat k terkecil sedemikian hingga G mempunyai partisi pembeda dengan k-anggota disebut **dimensi partisi** dari G dan dinotasikan dengan pd(G). Pada Tugas Akhir ini ditentukan **dimensi partisi** pada graf hasil korona Cm⊙ Kn dengan m, n bilangan bulat positif m ≥ 3, n ≥ 1. Dari analisis yang dilakukan diperoleh hasil bahwa **dimensi partisi** Cm& #8857;Kn, (Rumus) dengan p merupakan bilangan bulat positif terkecil yang memenuhi (pÂ¦n) ≥ m. Partisi pembeda; **Dimensi partisi**; Graf hasil korona.
1208100034	Graf adalah himpunan pasangan (V,E) dengan V adalah himpunan hingga tidak kosong dari simpul (vertex) dan E adalah himpunan sisi (edge), yaitu pasangan simpul dari V.Misalkan G=(V,E) adalah graf terhubung. Untuk setiap simpul v∈V(G) dan k-partisi terurut

	Π={S_1,S_2,â€¦,S_k} dari V(G), representasi dari v terhadap Π adalah k-vektor r(v│Π)=(d(v,S_1),d(v,S_2), â€¦,d(v,S_k)). Himpunan Π disebut partisi pembeda jika k-vektor r(v│Π) berbeda untuk setiap v∈V(G). Minimum k dari k-partisi pembeda dari V(G) disebut **dimensi partisi** dari G dan dinotasikan dengan pd(G). Graf serupa roda yang dibahas antara lain graf gir, helm, dan bunga matahari. Pada tugas akhir ini akan dilihat pengaruh penambahan anting terhadap dimensi partisi graf roda dan serupa roda. Dengan konsep himpunan pembeda yang dibahas sebelumnya, diperoleh dimensi partisi graf serupa roda dan graf serupa roda apabila diberi tambahan anting pada simpul-simpul tertentu pada graf untuk kemudian dilakukan analisis. Jumlah simpul (order) dari graf ditentukan dan terbatas karena tidak ada rumus tertentu untuk n buah simpul. Dari analisis yang dilakukan diperoleh hasil bahwa penambahan anting pada simpul-simpul graf roda dan serupa roda, kecuali simpul pusat, tidak mempengaruhi dimensi partisinya. partisi pembeda; **dimensi partisi**; graf roda; graf gir; graf helm; graf bunga matahari; penambahan anting
1208100024	Pewarnaan total graf G adalah fungsi yang memasangkan himpunan simpul dan himpunan sisi dengan himpunan bilangan asli yang merepresentasikan warna, sehingga tidak ada dua simpul atau dua sisi

	bersisihan memiliki warna yang sama. Pada umumnya, pewarnaan total hanya mewarnai elemen-elemen pada graf yang saling bersisihan dan melekat menggunakan warna yang berbeda. Kemudian muncul teori pewarnaan total dengan jumlah warna minimal. Jumlah warna minimal yang digunakan untuk mewarnai simpul dan sisi pada graf G disebut bilangan kromatik total G, dinotasikan dengan χ_T (G). Pada Tugas Akhir ini dikaji bilangan kromatik total pada graf bebas unichord adalah NP-lengkap dan **graf kincir** adalah ∆(G)+1. pewarnaan total; bilangan kromatik total; graf bebas unichord; **graf kincir**
1209100057	Sebuah graf G(V,E) dikatakan sebagai graf dengan n-coloring jika G dapat diwarnai dengan n warna sedemikian hingga tidak terdapat simpul-simpul saling bertetangga yang memiliki warna sama. Lebih lanjut, bila n menunjukkan jumlah minimum warna yang digunakan sehingga G tetap dapat diwarnai dan tidak terdapat simpul bertetangga dengan warna yang sama, maka n dikatakan sebagai bilangan kromatik dari G yang dinotasikan dengan χ(G). Dalam tugas Akhir ini dilakukan penentuan bilangan kromatik dari graf hasil amalgamasi dua buah graf. Operan yang digunakan dalam operasi amalgamasi ini berupa graf lengkap K_m dengan graf siklus C_n dan **graf kincir** W_m^k dengan W_n^l.

Proses pengklasteran dengan data yang berada pada Tabel 5.4 diawali dengan menggabungkan dokumen yang mempunyai NRP 1200109024 dengan 1206100047 dengan jarak (*similarity*) 0,863086 kemudian proses penggabungan selanjutnya yaitu pada dokumen yang mempunyai NRP 1206100015 dengan 1208100034 dengan jarak (*similarity*) 0,787087, proses selanjutanya *cluster* yang beranggotakan NRP 1200109024 dan 1206100047 bergabung dengan *cluster* yang beranggotakan NRP 1206100015 dan 1208100034 dengan jarak (similarity) 0,7345287. Sehingga terbentuk satu *cluster* yang beranggota 4. Setelah itu proses penggabungan terjadi pada dokumen yang mempunyai NRP 1208100024 dengan !208100057 dengan jarak (similarity) 0,428990. Tahap selanjutnya yaitu penggabungan *cluster* yang baru saja terbentuk dengan *cluster* yang mempunyai jumlah anggota 4 dengan jarak (similarity) 0,143004. Sehingga terbentuk *cluster* dengan jumlah anggota adalah 6 seperti pada Tabel 5.8

Setelah terbentuk menjadi satu *cluster*, ditentukan apa topik *cluster* tersebut. Topik didapatkan dari kata penting yang memiliki frekuensi kemunculan terbanyak pada *cluster*. Pada Tabel 5.4 dokumen abstrak pertama dan kedua membaha mengenai dimensi partisi dan graf kincir, sementara pada dokumen abstrak ketiga dan keempat membahas mengenai dimensi partisi saja, dan pada dokumen abstrak kelima dan keenam membahs mengenai graf kincir, dengan demikian jumlah frekunsi kemunculan dimensi partisi dengan graf kincir adalah sama sehingga topik/ pokok bahasan pada cluster ke 1 di departemen Matematika adalah **dimensi partisi atau graf kincir**

5.5 Evaluasi *Cluster*

Setelah proses pengklasteran menggunakan *Average Linkage Hierarchical Clustering* dan penentuan topik, langkah selanjutnya yaitu evaluasi *cluster*. Evaluasi *cluster* bertujuan untuk mengukur seberapa baik hasil *cluster* yang

dihasilkan. Pada penelitian ini digunakan *silhoutte coefficient*. Hasil perhitungan nilai *silhoutte coefficient* dapat bervariasi antara -1 hingga 1. Jika nilai *silhoutte coefficient* yang dihasilkan mendekati 1 maka *cluster* yang dihasilkan baik sementara jika mendekati -1 maka kurang baik. Data yang diolah pada tahap ini adalah data hasil *clustering* yang didapatkan dari proses sebelumnya. Berikut hasil *silhoutte coeffisient* di departemen Matematika, Fisika dan Teknik Perkapalan :

Tabel 5. 9 Nilai *silhoutte coefficient* di departemen Matematika

Nomer Urut	Jumlah Kata Penting	Jumlah Cluster	Nilai Silhoutte Coefficient
1	2	169	0.8384785392419544
2	4	116	0.5232464638370605
3	6	87	0.3481443473452224
4	7	68	0.2419374046373422
5	8	56	0.16488472875563812
6	10	31	0.015643005729301544
7	12	13	-0.08752570659932223
8	14	3	-0.11381558359274639
9	15	2	-0.1245800937203734
10	16	1	-1
11	17	1	-1

Tabel 5. 10 Nilai *silhoutte coefficient* di departemen Fisika

Nomer Urut	Jumlah Kata Penting	Jumlah Cluster	Nilai Silhoutte Coefficient
1	2	155	0.8069534671269435
2	4	99	0.4774776470523598
3	6	77	0.32708348429368744
4	7	70	0.28816951711863137
5	8	50	0.16101767099069228
6	10	23	0.030051691231973442
7	12	11	-0.04615453775172512
8	14	3	-0.09874247937811047
9	16	2	-0.11036538600441304
10	18	2	-0.13391053569582345
11	20	2	-0.1503313180590112
12	22	2	-0.16051791926897133
13	24	2	-0.17180397855878224
14	26	2	-0.17991566244064736
15	28	2	-0.18526577887315163
16	30	2	-0.18889680589674876
17	31	1	-1
18	32	1	-1

Tabel 5. 11 Nilai *silhoutte coefficient* di departemen Teknik Perkapalan

Nomer Urut	Jumlah Kata Penting	Jumlah Cluster	Nilai Silhoutte Coefficient
1	2	179	0.8220044755703022
2	4	127	0.5512346465578605
3	6	99	0.4257641710783688
4	8	37	0.11021285565982958
5	10	15	-0.0151217290436092
6	12	8	-0.0552779611613351
7	14	6	-0.0648787579102197
8	16	3	-0.0850094734216158
9	18	3	-0.0883593055559786
10	20	2	-0.0972381143749512
11	21	1	-1
12	22	1	-1

Hasil evaluasi *clustering* menunjukkan, pada departemen Matematika nilai *silhoutte coefficient* terbaik adalah 0.8384785392419544 dengan jumlah *cluster* adalah 169 dan jumlah kata penting adalah 2. Sementara pada departemen Fisika nilai *silhoutte coefficient* terbaik adalah 0.8069534671269435, dengan jumlah kata penting adalah 2 dan jumlah *cluster* adalah 155. Nilai *silhoutte coefficient* terbaik yang dihasilkan pada departemen Teknik Perkapalan yaitu 0.7471874459147052 dengan jumlah kata penting adalah 2 dan jumlah *cluster* adalah 179.

Berdasarkan Tabel 5.9, Tabel 5.10 dan Tabel 5.11, menunjukkan jumlah *cluster* akan mempengaruhi nilai *silhoutte coefficient*. Jumlah *cluster* yang semakin banyak akan menghasilkan nilai *silhoutte coefficient* yang semakin tinggi. Hal ini dikarenakan ketika jumlah *cluster* yang dihasilkan banyak, maka jumlah anggota pada setiap *cluster* akan lebih sedikit, hal ini menyebabkan setiap anggota

mempunyai kedekatan kemiripan topik yang tinggi dengan anggota yang lain.

BAB VI
PENUTUP

Pada bab ini berisi tentang beberapa kesimpulan yang dihasilkan berdasarkan penelitian yang telah dilaksanakan dan saran yang dapat digunakan jika penelitian ini dikembangkan.

6.1 Kesimpulan

Berdasarkan analisis terhadap hasil pengujian program, maka dapat diambil kesimpulan sebagai berikut :

1. Penentuan nilai parameter jumlah kata penting pada proses ekstraksi kandidat kata penting dengan *Rapid Automatic Keyphrase Extraction* (RAKE) menghasilkan kata penting dalam abstrak laporan Tugas Akhir. Semakin tinggi nilai parameter yang dihasilkan RAKE maka semakin banyak total kata penting yang dihasilkan.
2. Banyaknya *cluster* yang dihasilkan dengan metode *Average Linkage Hierarchical Clustering* tergantung dari jumlah parameter kata penting yang dimasukkan pada proses RAKE. Semakin banyak jumlah kata penting yang dimasukkan maka semakin sedikit hasil *cluster* yang dihasilkan. Jumlah *cluster* mempengaruhi nilai *silhoutte coefficient*. Semakin banyak jumlah *cluster*, nilai *silhoutte coefficient* yang dihasilkan semakin tinggi. Pengklasteran terbaik mempunyai nilai *silhoutte coefficient* paling tinggi. Dalam uji coba penelitian ini, pengklasteran terbaik dihasilkan dengan memasukkan 2 kata penting.

6.2 Saran

Saran yang dapat diberikan untuk pengebangan penelitian selanjutnya adalah

1. Perlu adanya penambahan *stopword* yang sesuai dengan data agar hasil kata penting yang dihasilkan lebih baik.

2. Pada metode *Rapid Automatic Keyphrase Extraction* (RAKE) tidak ada proses *stemming*. Oleh karena itu perlu adanya proses *stemming* sehingga kata penting yang dihasilkan akan lebih baik.
3. Proses pengklasteran terhadap abstrak dapat menggunakan metode lainnya yang menyajikan hasil *cluster* lebih baik.

DAFTAR PUSTAKA

[1] (2015), **Panduan Akademik Program Sarjana dan Magister Matematika 2014-2019**, Fakultas Matematika dan Ilmu Pengetahuan Alam, Teknologi Sepuluh Nopember, Surabaya.

[2] Purnama, A., (2011), www.kompasiana.com diakses pada tanggal 7 Februari 2017 pukul 09.27

[3] Rosyid, N.M., (2009), **Penentuan Kemiripan Topik Proyek Akhir berdasarkan Abstrak pada Jurusan Teknik Informatika menggunakan metode Single Linkage Hierarchical**, Institut Teknologi Sepuluh Nopember (ITS). Surabaya.

[4] Subandi, N.A, (2014), **Clustering Dokumen Skripsi Berdasarkan Abstrak dengan Menggunakan Bisecting K-Means, Institut Pertanian Bogor**, Bogor.

[5] Alfina, T., Santosa, B., Barakbah, A.R., (2012), **Analisa Perbandingan Metode *Hierarchical Clustering, K-Means* dan Gabungan Keduanya dalam *Cluster* Data (Studi kasus : Problem Kerja Praktek Jurusan Yeknik Industri ITS),** Institut Teknologi Sepuluh Nopember (ITS), Surabaya.

[6] Laeli, S., (2014), **Analisis *Cluster* dengan *Average Linkage Method* dan *Ward's Method* untuk Data Responden Nasabah Asuransi Jiwa Unit Link**, Universitas Negeri Yogyakarta, Yogyakarta.

[7] Andini, S., (2013), **Klasifikasi Dokumentasi Teks Menggunakan Algoritma Naive Bayes dengan Bahasa Pemrograman Java**, Jurnal Teknologi Informasi & Pendidikan, Vol.6 No.2 September 2013.

[8] Nugroho. E., (2011), **Perancangan Sistem Deteksi Plagiarisme Dokumen Teks Dengan Menggunakan Algoritma Rabin-Karp**, Program Studi Ilmu Komputer, Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Brawijaya.

[9] Rose, S., Engel, D., Cramer, N. dan Cowley, W., (2010), ***Automatic Keyword Extraction from Individual Documents***, *Text Mining: Applications and Theory*.

[10] Berry, M.W., dan Kogan, J., (2010), **Text Mining: Applications and Theory**, John Wiley & Sons.

[11] Ulinnuha, N., Ginardi, R.V.H, Purwitasari, D., (2013), **Ekstraksi Kata Kunci menggunakan M-RAKE pada Dokumen Penelitian**, Institut Teknologi Sepuluh Nopember, Surabaya.

[12] Rukmi, A.M., dan Iqbal, I.M, (2014), **Kajian Graf Analisis Jejaring Sosial Pada Pengklasteran Penelitian**, Institut Teknologi Sepuluh Nopember, Surabaya.

[13] Rammawati, L., Sihwi, S.W, dan Suryani, E., (2015), **Analisa Clustering menggunakan Metode K-Means dan Hierarchical Clustering (Studi Kasus : Dokumen Skripsi Jurusan Kimia, FMIPA, Universitas Sebelas Maret),** Universitas Sebelas Maret, Surakarta.

[14] Lee, D.L., (1997), ***Document Ranking and the Vector-Space Model***. Hong Kong University of Science and Technology. Hongkong.

[15] Andayani, S., (2007), **Pembentukan *Cluster* dalam Knowledge Discovery in Database dengan Algoritma K-Means**, Seminar Nasional Matematika dan Pendidikan Matematika 2007, Universitas Negeri Yogyakarta, Yogyakarta.

[16] Risal, H., (2006), **Klasterisasi Dokumen XML berdasarkan strukturnya dengan menggunakan Algoritma berbasis Hirarki**, Teknik Informatika, Institut Teknolgi Sepuluh Nopember.

[17] Luthfiarta, A., Zeniarja, J., dan Salam, A., (2013), **Algoritma Latent Semantic Analysis (LSA) pada Peringkas Dokumen Otomatis untuk Proses Clustering Dokumen**, Seminar Nasional Teknologi

Informasi dan Komunikasi Terapan 2013, ISSBN : 979-26-0266-6, Universitas Dian Nuswantoro, Semarang.

[18] Han, J., Kamber, M., dan Pei, J., (2012), ***Data Mining Concepts and Techniques Third Edition***, USA.

[19] Wahyuni, I., Anliya, Y.A, Mahmudy, W.F., (2016), **Clustering Nasabah Bank Berdasarkan Tingkat Likuiditas Menggunakan *Hybird* Particle Swarm Optimization dengan K-Means**, Jurnal Ilmiah Teknologi dan Informasi ASIA (JITIKA), Vol.10, No.2, Agustus 2016, Universitas Brawijaya, Malang.

LAMPIRAN A
Hasil *Cluster*

a. Departemen Fisika dengan memasukkan 2 kata penting

Cluster 16	
NRP	:1107100006, 1107100046, 1107100032, 1108100701,
Topik	:sistem potensial listrik
Cluster 17	
NRP	:1107100009, 1108100056,
Topik	:alat ukur daya isolasi
Cluster 18	
NRP	:1107100011, 1108100027, 1108100008, 1108100017, 1108100023, 1109100005,
Topik	:dye sensitized solar cell
Cluster 19	
NRP	:1107100013,
Topik	:merubah domain data seismik
Cluster 20	
NRP	:1107100014, 1108100704,
Topik	:nilai absorbansinya
Cluster 21	
NRP	:1107100015,
Topik	:disimpulkan elemen thermoelektrik tipe tec
Cluster 22	
NRP	:1107100016, 1108100011, 1108100070,
Topik	:voltage standing wave ratio
Cluster 23	
NRP	:1107100018,
Topik	:plasma enhanced chemical vapor deposition
Cluster 24	
NRP	:1107100019, 1108100039, 1109100014, 1107100054,
Topik	:metode kopresipitasi sederhana
Cluster 25	
NRP	:1107100020,
Topik	:informasi visualisasi nilai intensitas dimensi

b. Departemen Matematika dengan memasukkan 2 kata penting

Cluster 1	
NRP	:1200109024, 1206100047, 1206100015, 1208100034, 1208100024, 1209100057,
Topik	:dimensi partisi/ graf kincir
Cluster 2	
NRP	:1202100005, 1207100066,
Topik	:multivariate
Cluster 3	
NRP	:1204100019,
Topik	:backward heat equation
Cluster 4	
NRP	:1204100040,
Topik	:highly active antiretroviral theraphy
Cluster 5	
NRP	:1205100022,
Topik	:pendekatan hiperbolisasi histogram fuzzy intuisi atanassov
Cluster 6	
NRP	:1205100028,
Topik	:kebijakan keamanan
Cluster 7	
NRP	:1205100065,
Topik	:input sistem pakar fuzzy
Cluster 8	
NRP	:1205100068, 1206100705,
Topik	:advanced encryption standard
Cluster 9	
NRP	:1206100032, 1206100709, 1206100710, 1208100054, 1206100060,
Topik	:metode ensemble kalman filter
Cluster 10	
NRP	:1206100034, 1208100009,
Topik	:probabilitas server menganggur

c. Departemen Teknik Perkapalan dengan memasukkan 2 kata penting

Cluster 1	
NRP	:4103100039,
Topik	:dua unit kapal tanker pertamina
Cluster 2	
NRP	:4103100041,
Topik	:menganalisa proses evakuasi
Cluster 3	
NRP	:4103109013,
Topik	:dwt pal surabaya divisi kapal niaga terdapatsedikit
Cluster 4	
NRP	:4104100003, 4107100031, 4104100028, 4107100065, 4107100086, 4105100017, 4108100011, 4108100066, 4109100023, 4109100014, 4109100031, 4108100012, 4109100074, 4109100068,
Topik	:proses pengelasan
Cluster 5	
NRP	:4104100025, 4107100045, 4107100079,
Topik	:teori hull girder respon analysis
Cluster 6	
NRP	:4104100036, 4110100025, 4107100027, 4106100005, 4108100007,
Topik	:computational fluid dynamics

d. Departemen Fisika dengan memasukkan 4 kata penting

Cluster 1	
NRP	:1105100022, 1106100007, 1110100002, 1107100030, 1108100058,
Topik	:mikrokontroler atmega
Cluster 2	
NRP	:1105100026, 1107100024, 1108100003, 1108100013, 1108100015, 1108100025, 1108100043,

1108100016, 1108100040, 1109100050, 1107100062,
1108100032, 1108100054, 1108100055, 1108100064,
1106100017, 1109100008, 1109100031, 1109100046,
1110100008, 1110100054, 1110100026, 1110100034,
1110100043, 1106100026, 1110100003, 1108100010,
1108100026, 1109100002, 1110100029, 1106100051,
1107100055, 1109100038, 1107100027, 1110100059,
1107100019, 1108100039, 1109100014, 1107100054,
Topik :scanning electron microscopy
Cluster 3
NRP :1105100032,
Topik :litologi batuan karbonat
Cluster 4
NRP :1106100005, 1107100006, 1107100046,
1108100063, 1107100032, 1108100701,
Topik :syarat batas
Cluster 5
NRP :1106100009, 1107100002, 1107100001,
1107100025,
Topik :sumber gelombang akustik
Cluster 6
NRP :1106100018, 1109100001,
Topik :metode reaksi kimia
Cluster 7
NRP :1106100038,
Topik :data gain reflektor jenis
Cluster 8
NRP :1106100053,
Topik :alat magnetometer proton envi scintrex/
Cluster 9
NRP :1106100063,
Topik :fasa spinel mgalo
Cluster 10
NRP :1107100004,
Topik :cartesian coordinates
Cluster 11
NRP :1107100009, 1108100056,
Topik :alat ukur daya isolasi

Cluster 12
NRP :1107100011, 1108100006, 1108100027, 1107100063, 1109100030, 1108100008, 1108100017, 1108100023, 1109100005, 1107100014, 1108100704, 1107100050, 1109100003, 1109100060, 1109100034, 1109100704, 1108100047, 1108100050, 1109100035, 1108100041, 1109100054, 1109100025, 1110100024, 1110100039,
Topik :serat optik

e. Departemen Matematika dengan memasukkan 4 kata penting

Cluster 1
NRP :1200109024, 1206100015, 1206100047, 1208100034, 1209100057, 1208100024, 1209100053,
Topik :dimensi partisi/ graf kinci
Cluster 2
NRP :1202100005, 1207100066, 1209100038, 1210100028,
Topik :ekonomi jawa timur/ multivariate/ produk domestik regional bruto
Cluster 3
NRP :1204100019, 1207100017, 1210100039, 1206100701, 1208100059, 1208100072, 1204100040, 1207100046, 1207100037, 1208100042, 1207100702, 1207100056, 1207100045, 1208100035, 1208100033, 1207100047, 1210100056, 1210100045, 1209100002, 1207100706, 1208100070, 1209100082, 1209100088, 1209100050, 1209100054, 1209100703, 1210100069, 1210100072, 1208100058, 1209100092, 1209100070, 1209100079, 1208100703,
Topik :metode beda hingga
Cluster 4
NRP :1205100022, 1209100033,
Topik :citra digital berwarna

Cluster 5
NRP :1205100028,
Topik :kebijakan keamanan
Cluster 6
NRP :1205100065,
Topik :input sistem pakar fuzzy
Cluster 7
NRP :1205100068, 1206100705, 1209100027,
Topik :advanced encryption standard/ dua proses utama
Cluster 8
NRP :1206100032, 1206100709, 1208100054, 1206100710, 1206100042, 1208100705, 1208100044, 1209100704, 1206100060, 1206100719, 1208100707, 1207100063, 1209100028,
Topik :ensemble kalman filter
Cluster 9
NRP :1206100034, 1208100009,
Topik :probabilitas server menganggur
Cluster 10
NRP :1206100043,
Topik :metode resampling bootstrap

f. Departemen Teknik Perkapalan dengan memasukkan 4 kata penting

Cluster 1
NRP :4103100039, 4104100003, 4107100037, 4109100086, 4109100702, 4108100111, 4108100113, 4108100026, 4109100058, 4108100017, 4108100101, 4108100031, 4108100036, 4107100009, 4107100069, 4109100072, 4107100046, 4105100060, 4106100045, 4106100064, 4105100017, 4108100011, 4108100066, 4109100023, 4109100014, 4109100031, 4108100012, 4109100074, 4109100068, 4107100086, 4107100031, 4109100034, 4105100027, 4109100043, 4107100096, 4106100059, 4107100007, 4105100066, 4106100074,

4107100026, 4107100022, 4108100029, 4108100084, 4108100059, 4109100087, 4104100025, 4106100089, 4104100028, 4105100045, 4107100045, 4107100079, 4108100038, 4105100072, 4107100011, 4108100063, 4108100094, 4107100065, 4108100098, 4107100005, 4104100036, 4110100025, 4108100007, 4106100005, 4105100019, 4105100046, 4107100027, 4107100095, 4108100058, 4109100055, 4108100039, 4107100010, 4107100041,
Topik :metode elemen hingga
Cluster 2
NRP :4103100041,
Topik :analisa evakuasi sederhana
Cluster 3
NRP :4103109013,
Topik :dwt pal surabaya divisi kapal niaga terdapatsedikit
Cluster 4
NRP :4104100049,
Topik :memvariasi tongkang
Cluster 5
NRP :4104100061,
Topik :biaya transportasi distribusi muatan ekspor
Cluster 6
NRP :4104100072,
Topik :kluster industri jawa barat
Cluster 7
NRP :4105100011,
Topik :building berth bertipe end launching
Cluster 8
NRP :4105100020, 4106100024,
Topik :kapal patroli

g. Departemen Fisika dengan memasukkan 6 kata penting

Cluster 1
NRP :1105100022, 1106100007, 1110100002, 1107100030, 1107100050, 1109100034, 1108100058, 1109100003, 1109100060, 1109100704, 1108100050, 1109100035, 1108100041, 1109100054, 1108100047, 1110100039, 1107100014, 1108100704, 1108100065, 1110100024, 1109100025, 1109100061, 1107100011, 1108100006, 1108100027, 1108100008, 1108100023, 1108100017, 1109100005, 1108100014, 1107100063, 1109100030,
Topik :serat optik
Cluster 2
NRP :1105100026, 1108100003, 1108100013, 1108100016, 1108100043, 1108100015, 1108100025, 1108100055, 1108100064, 1109100050, 1107100019, 1108100039, 1109100014, 1107100054, 1107100051, 1107100022, 1107100024, 1107100044, 1110100026, 1107100062, 1108100032, 1108100054, 1106100026, 1110100003, 1110100034, 1110100059, 1110100043, 1109100002, 1110100029, 1108100010, 1108100026, 1106100051, 1107100027, 1107100055, 1109100038, 1108100040, 1106100017, 1109100031, 1109100046, 1109100008, 1109100032, 1108100061, 1110100054, 1110100008, 1110100009, 1107100045,
Topik :scanning electron microscopy
Cluster 3
NRP :1105100032,
Topik :litologi batuan karbonat
Cluster 4
NRP :1106100005, 1107100006, 1107100046, 1107100032, 1108100701, 1107100049, 1107100058, 1108100051, 1110100062, 1108100063,
Topik :data sumur/ syarat batas

Cluster 5
NRP :1106100009, 1107100002, 1107100001, 1107100025,
Topik :function generator/ sumber gelombang akustik
Cluster 6
NRP :1106100018, 1109100001,
Topik :metode reaksi kimia
Cluster 7
NRP :1106100038,
Topik :bidang pencahayaan
Cluster 8
NRP :1106100053,
Topik :alat magnetometer proton envi scintrex/

h. Departemen Matematika dengan memasukkan 6 kata penting

Cluster 1
NRP :1200109024, 1206100015, 1206100047, 1208100034, 1208100024, 1209100053, 1207100001, 1207100069, 1209100057, 1209100069, 1205100028, 1207100033, 1208100029, 1208100046, 1209100056, 1204100040, 1207100046, 1207100702, 1208100033, 1208100042, 1207100037, 1207100045, 1208100035, 1207100056, 1209100066, 1204100019, 1207100017, 1210100039, 1206100701, 1208100059, 1208100072, 1206100034, 1208100009, 1209100094, 1209100050, 1209100054, 1209100703, 1210100069, 1210100072, 1207100047, 1210100056, 1210100045, 1209100002, 1207100706, 1208100070, 1209100082, 1209100088, 1208100058, 1209100092, 1209100070, 1209100079, 1208100703,
Topik :metode beda hingga

Cluster 2
NRP :1202100005, 1207100066, 1210100028, 1209100038,
Topik :ekonomi jawa timur/ multivariate/ produk domestik regional bruto
Cluster 3
NRP :1205100022, 1209100033, 1208100002, 1209100076, 1208100031,
Topik :ilmu pengetahuan
Cluster 4
NRP :1205100065, 1209100073, 1209100089, 1210100005,
Topik :penerapan logika fuzzy/ sistem pakar fuzzy
Cluster 5
NRP :1205100068, 1209100027, 1206100705,
Topik :advanced encryption standard/ dua proses utama
Cluster 6
NRP :1206100032, 1208100054, 1206100709, 1206100710, 1206100042, 1208100044, 1208100705, 1209100704, 1206100060, 1206100719, 1207100063, 1208100707, 1209100028,
Topik :ensemble kalman filter

i. Departemen Teknik Perkapalan dengan memasukkan 6 kata penting

Cluster 1
NRP :4103100039, 4108100080, 4108100089, 4109100085, 4104100003, 4107100037, 4109100086, 4109100702, 4108100111, 4109100058, 4108100113, 4108100017, 4108100101, 4108100031, 4108100036, 4107100009, 4107100069, 4107100046, 4109100072, 4105100060, 4106100045, 4106100064, 4108100040,

4105100027, 4109100043, 4107100096, 4106100059,
4107100007, 4105100066, 4106100074, 4107100014,
4107100026, 4107100103, 4110100045, 4108100102,
4106100025, 4110100053, 4109100060, 4108100026,
4107100022, 4108100029, 4108100084, 4108100043,
4108100087, 4104100025, 4107100045, 4107100079,
4108100038, 4105100045, 4107100011, 4105100072,
4107100065, 4108100063, 4108100094, 4104100028,
4108100098, 4107100005, 4107100095, 4105100019,
4105100046, 4108100058, 4109100055, 4108100039,
4106100089, 4107100010, 4107100041, 4104100061,
4108100048, 4109100006, 4109100011, 4109100009,
4105100017, 4108100011, 4108100012, 4108100066,
4109100023, 4109100014, 4109100031, 4109100074,
4109100068, 4107100086, 4107100031, 4109100034,
4110100099, 4104100036, 4106100042, 4107100027,
4108100007, 4110100025, 4106100005, 4108100059,
4109100087, 4108100002, 4108100090, 4108100093,
4110100039, 4110100065,

Topik :metode elemen hingga

Cluster 2

NRP :4103100041,

Topik :analisa evakuasi sederhana

Cluster 3

NRP :4103109013,

Topik :analisdeskriptif menyimpulkan

Cluster 4

NRP :4104100049,

Topik :memvariasi tongkang

Cluster 5

NRP :4104100072, 4105100058, 4107100015,
4110100052,

Topik :biaya transportasi

Cluster 6	
NRP	:4105100011, 4109100701,
Topik	:pendapatan bersih pertahun
Cluster 7	
NRP	:4105100020, 4106100024,
Topik	:kapal patroli

LAMPIRAN B
Source code

Preprocess

```
     System.out.println(jComboBox1.getSelectedItem());
        int jur = 0;
        int thn= 0;
        if(jComboBox1.getSelectedItem().equals("Jurusan
Fisika")){
            jur = 11;
        }
        else
if(jComboBox1.getSelectedItem().equals("Jurusan
Matematika")){
            jur = 12;
        }
        else
if(jComboBox1.getSelectedItem().equals("Jurusan
Statistika")){
            jur = 13;
        }
        else
if(jComboBox1.getSelectedItem().equals("Jurusan
Kimia")){
            jur = 14;
        }
        else
if(jComboBox1.getSelectedItem().equals("Jurusan
Biologi")){
            jur = 15;
        }
        else
if(jComboBox1.getSelectedItem().equals("Jurusan Teknik
Mesin")){
            jur = 21;
        }
        else
if(jComboBox1.getSelectedItem().equals("Jurusan Teknik
Elektro")){
            jur = 22;
        }
        else
if(jComboBox1.getSelectedItem().equals("Jurusan Teknik
Kimia")){
            jur = 23;
        }
```

```
        else
if(jComboBox1.getSelectedItem().equals("Jurusan Teknik
Fisika")){
            jur = 24;
        }
        else
if(jComboBox1.getSelectedItem().equals("Jurusan Teknik
Industri")){
            jur = 25;
        }
        else
if(jComboBox1.getSelectedItem().equals("Jurusan Teknik
Material dan Metalurgi")){
            jur = 27;
        }
        else
if(jComboBox1.getSelectedItem().equals("Jurusan Teknik
Sipil")){
            jur = 31;
        }
        else
if(jComboBox1.getSelectedItem().equals("Jurusan
Arsitektur")){
            jur = 32;
        }
        else
if(jComboBox1.getSelectedItem().equals("Jurusan Teknik
Lingkungan")){
            jur = 33;
        }
        else
if(jComboBox1.getSelectedItem().equals("Jurusan Desain
Produk Industri")){
            jur = 34;
        }
        else
if(jComboBox1.getSelectedItem().equals("Jurusan Teknik
Geomatika")){
            jur = 35;
        }
        else
if(jComboBox1.getSelectedItem().equals("Jurusan
Perancangan Wilayah dan Kota")){
            jur = 36;
        }
        else
if(jComboBox1.getSelectedItem().equals("Jurusan Teknik
Perkapalan")){
            jur = 41;
```

```
        }
        else
if(jComboBox1.getSelectedItem().equals("Jurusan Teknik
Sistem Perkapalan")){
            jur = 42;
        }
        else
if(jComboBox1.getSelectedItem().equals("Jurusan Teknik
Kelautan")){
            jur = 43;
        }
        else
if(jComboBox1.getSelectedItem().equals("Jurusan Teknik
Informatika")){
            jur = 51;
        }
        else
if(jComboBox1.getSelectedItem().equals("Jurusan Sistem
Informasi")){
            jur = 52;
        }
        else {jur = 99;}


if(jComboBox3.getSelectedItem().equals("2012")){
            thn = 2012;
        }
        else
if(jComboBox3.getSelectedItem().equals("2013")){
            thn = 2013;
        }
        else
if(jComboBox3.getSelectedItem().equals("2014")){
            thn = 2014;
        }
        else
if(jComboBox3.getSelectedItem().equals("2015")){
            thn = 2015;
        }
        else
if(jComboBox3.getSelectedItem().equals("2016")){
            thn = 2016;
        }
        else{thn = 99;}
        System.out.println(thn);
        databaseTA db = new databaseTA();
        RAKE rake = new RAKE();
        try {
             // TODO add your handling code here:
```

```
                db.connectFirst();
                db.executeUpdate("DELETE FROM
representasi_dok_rake");
                DokumenTA dt = new DokumenTA();
                dt.in_NRP_TAHUN(jur,thn);
                LinkedList<DokumenTA> listTA =
dt.readPartDokumen();
                System.out.println(listTA.size());
                DefaultTableModel dtm=(DefaultTableModel)
jTable1.getModel();
                dtm.setRowCount(0);
                String [][]a=dt.getData();
                String []b = new String[4];
                for(int k=0;k<listTA.size();k++){
                    b[0]=a[0][k];
                    b[1]=a[1][k];
                    b[2]=a[2][k];
                    b[3]=a[3][k];
                    dtm.addRow(b);
                }
                // memuculkan jumlah data

jmldata.setText(String.valueOf(listTA.size()));

                for(DokumenTA d : listTA){
                    System.out.println(d.getId());
                    rake.formRepDokumen(d);
                }
                db.destroyConnection();
            } catch (SQLException ex) {

Logger.getLogger(TA.class.getName()).log(Level.SEVERE,
null, ex);

        }
        }
```

Ekstraksi

```
        Integer jumkp =
Integer.parseInt(jumKPtext.getText());
        boolean rakeStat = true;
       }
        System.out.println(rakeStat);

            try{
                rake.prosesRAKE2(jumkp);
            } catch(SQLException ex){

Logger.getLogger(TA.class.getName()).log(Level.SEVERE,
null, ex);
            }

            databaseTA db = new databaseTA();
            try {
            db.connectFirst();
            ResultSet rs = db.executeSelect("SELECT *\n"
+
                                                    "FROM
`list_kata_penting_norake`\n" +
                                                    "LIMIT
0, 30");
            while(rs.next()){
                String[]s ={rs.getString(1)};
                dtmkk.addRow(s);
            }
            rs = db.executeSelect("SELECT *\n" +
                                  "FROM
`list_kata_penting_norake`\n");
            int jum = 0;
            while(rs.next()){
                jum += 1;
            }
            konseptabel.setModel(dtmkk);
            jumlistKP.setText(String.valueOf(jum));
            } catch (SQLException ex){

Logger.getLogger(TA.class.getName()).log(Level.SEVERE,
null, ex);
            }
            LSA lsa = new LSA();

            try{
               inputaverage = lsa.prosesLSA();
            } catch (SQLException ex){
```

```
Logger.getLogger(TA.class.getName()).log(Level.SEVERE,
null, ex);
            } catch (FileNotFoundException ex){

Logger.getLogger(TA.class.getName()).log(Level.SEVERE,
null, ex);
            }
            this.data = lsa.getData();
            this.NRP = lsa.getNrp();
            this.kataKunci = lsa.getKataKunci();
```

Average Linkage

```
    String[] nrpasli = this.NRP;

       for(String a:nrpasli)
       System.out.println(a);
       averagelinkage coba = new
averagelinkage(this.data, nrpasli, this.kataKunci);
       String[] hasil = coba.showCluster();
      // String[] hasil = coba.showCluster(nrpasli);
       coba.showKataKunci(hasil);

       nrpCluster = coba.getNrpCluster();

    // coba.hitungSilhoutte();

       // Menampilkan Tabel klaster
     /*  DefaultTableModel tabel =(DefaultTableModel)
tabel_klaster.getModel();
              dtm.setRowCount(0);
              String [][]a=dt.getData();
              String []b = new String[4];
              for(int k=0;k<listTA.size();k++){
                  b[0]=a[0][k];
                  b[1]=a[1][k];
                  b[2]=a[2][k];
                  b[3]=a[3][k];
                  dtm.addRow(b);
              }
       */
       int[] jumang=coba.getJumlahNRP();
       String[] top=coba.getTopik();
       char[] temp;
       for(int e=0; e<hasil.length; e++){
           temp=hasil[e].toCharArray();
```

```
            for(int d=0;d<temp.length;d++){
                if (temp[d]=='/'){
                    temp[d]=',';
                }
            }
            hasil[e]=String.valueOf(temp);
       }



       for(int i=0; i<coba.getJumlahNRP().length; i++){
          tabel_klaster.setValueAt(i+1,i,0);

tabel_klaster.setValueAt(String.valueOf(jumang[i]),i,1)
;
          tabel_klaster.setValueAt(hasil[i],i,2);
          tabel_klaster.setValueAt(top[i],i,3);

       }


       JOptionPane.showMessageDialog(this, "Proses
Clustering Selesai");        // TODO add your handling
code here:
    }
```

Silhoutte Coefficient

```
        silhoutte saya = new silhoutte(data, NRP,
nrpCluster);

        totalCluster = saya.getjumlahCluster();
        siluet = saya.gettotalSiluet();
        jumklas.setText(String.valueOf(totalCluster));
        nilaisil.setText(String.valueOf(siluet));
```

BIODATA PENULIS

Penulis bernama lengkap **Mega Fatmawati**, lahir di Banyuwangi, 15 Juni 1995. Anak kedua dari pasangan Hariyadi dan Sri suati, serta memiliki kakak laki-laki Adi Purwanto. Penulis mengikuti pendidikan dasar Sekolah Dasar di Kota Banyuwangi dilanjutkan Sekolah Menengah Pertama dan Sekolah Menegah Atas di Kota Gresik. Penulis menempuh pendidikan di SD Negeri 4 Purwoharjo, SMP Muhammadiyah 1 Gresik, dan SMA Negeri 1 Gresik. Setelah Lulus dari SMAN 1 Gresik pada tahun 2013 yang lalu, penulis melanjutkan pendidikan tingginya di Institut Teknologi Sepeluh Nopember (ITS) Surabaya dengan mengambil Jurusan Matematika dengan bidang minat Ilmu Komputer. Selama mengikuti perkuliahan di ITS, penulis turut aktif dalam beberapa kegiatan kemahasiswaan sebagai Asisten Direktur Bidang Bisnis periode 2014/2015, Ketua Devisi Pengabdian Masyarakat periode 2015/2016. Informasi lebih lanjut mengenai Tugas Akhir ini dapat ditujukan ke penulis melalui email: mega.fatmawati15@gmail.com.